

UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et de génie informatique

Développement d'une nouvelle approche d'essais pour
l'évaluation de systèmes avancés d'assistance et d'aide à la
conduite (ADAS) dans les véhicules intelligents ou hautement
automatisés sous de multiples conditions

Thèse de doctorat

Spécialité : génie électrique et informatique

par

Nacer Eddine Chelbi

Sherbrooke (Québec) Canada

27 Mai 2019

Membres du jury

Denis GINGRAS Dr. Ing.

Professeur à l'université de Sherbrooke et directeur du laboratoire sur l'intelligence
véhiculaire (LIV) de la faculté de génie, Université de Sherbrooke

Directeur

Joao Pedro Fernandes Trovao

Professeur à la faculté de génie, Université de Sherbrooke

Rapporteur

Eric Plourde PhD

Professeur à la faculté de génie, Université de Sherbrooke

Évaluateur

Arnaud de La Fortelle

Professeur à MINES ParisTech - université PSL et directeur du Centre de Robotique

Évaluateur externe

Claude SAUVAGEAU Ing.

Directeur essais et immobilisations à PMG Technologies Inc.

Évaluateur externe représentant du partenaire industriel

Résumé

Les systèmes avancés d'aide à la conduite (ADAS) doivent passer des tests intensifs avant d'être mis en production, mais les tests avec des véhicules réels prennent beaucoup de temps, sont coûteux, difficiles à reproduire et présentent des risques. Des essais avec des véhicules réels seront toujours nécessaires à l'avenir, mais ne seront pas suffisants pour répondre à la totalité des exigences de fiabilité et de sécurité. Pour démontrer les performances attendues d'un système ADAS dans un véhicule intelligent ou autonome, les méthodes d'évaluation doivent inclure des simulations, des essais sur piste et des essais sur route [25]. Les véhicules intelligents et autonomes continueront à rendre la conduite plus facile et plus sécuritaire. Néanmoins, la question ultime reste à savoir quelle est l'approche d'évaluation optimale à proposer, qui aura la capacité de valider le comportement et les performances attendues des systèmes embarqués dans des véhicules intelligents et autonomes pendant tous les stades: développement, démonstration et déploiement. À notre connaissance, la plupart des études / propositions qui tentent de combiner plusieurs approches (2 ou 3) sont appliquées pendant le stade de développement.

Cette thèse présente une nouvelle approche d'évaluation des systèmes avancés d'aide à la conduite dans un véhicule intelligent ou hautement automatisé conduit par une entité externe pendant la phase de déploiement. Cette approche permet à la fois d'identifier un ensemble de pires scénarios pour une application ADAS donnée et de combiner les trois approches d'évaluation mentionnées précédemment.

Pour ce faire, cette étude propose plusieurs solutions qui sont regroupées en trois parties. La première « Synthèse de scénarios, stratégie d'échantillonnage et simulations » comprend une description du système ADAS à évaluer et de ses différents critères d'évaluation, propose une synthèse des scénarios de tests les plus pertinents avec les paramètres de fonctionnement de chaque scénario. Ensuite, nous traitons la base de données FOT en implémentant une stratégie d'échantillonnage appropriée et à la fin de cette partie, des tests virtuels sont mis en oeuvre dans un environnement de simulation véhiculaire.

La deuxième partie « Évaluation et classification des risques » se concentre sur la collecte des résultats de simulation, puis sur l'évaluation et la classification du risque de chaque test, ce qui

nous permet ensuite de récupérer les niveaux de risques et d'avoir une estimation approximative de l'ensemble de scénarios défavorables.

La troisième partie « Validation » traite les résultats des essais sur piste de l'ADAS étudié et des différentes techniques d'apprentissage automatique et ensembliste utilisées pour créer son modèle prédictif. Ensuite, le traitement de la base de données FOT et l'implémentation d'une stratégie d'échantillonnage plus avancée et à la fin la collecte des résultats de prédiction, puis sur l'évaluation de risque de chaque test et sur sa classification à l'aide d'une technique de classification non supervisée, ce qui nous permet de construire et sélectionner finalement un ensemble des pires scénarios.

Mots-clefs : Systèmes Avancés d'Aide à la Conduite (ADAS); Apprentissage Automatique et Ensembliste; Évaluation Et Validation; Véhicules Intelligents et Autonomes; Pires Scénarios (Worst-Case Scenarios); Méthodes Monte-Carlo par Chaînes de Markov (MCMC); Algorithme de Metropolis-Hastings, Tests Operationnels sur Terrain (FOT)

Remerciements

"No two things have been combined better than knowledge and patience"

Je voudrais tout d'abord adresser mes sincères remerciements à mon directeur de recherche, le professeur Denis Gingras, pour ses précieux conseils et sa compréhension, et pour avoir orienté et encouragé mes travaux de recherche.

Je voudrais aussi remercier monsieur Claude Sauvageau pour ses conseils et ses commentaires, et le professeur Joao Pedro Fernandes Trovao pour avoir accepté d'être rapporteur pour ma thèse, ainsi que le professeur Eric Plourde et le professeur Arnaud de la Fortelle pour avoir accepté de faire partie de mon jury de thèse.

Je remercie aussi tous mes collègues et partenaires de travail pour les discussions constructives et pour les bons moments que nous avons eu durant ces années d'études.

Je tiens à remercier notre partenaire industriel PMG Technologies et Transport Canada, le programme de financement «MITACs Accelerate» du Canada, le Conseil national de recherches en sciences et en génie (CRSNG) du Canada pour avoir soutenu mon projet de recherche.

Je souhaiterais enfin remercier tout particulièrement mes parents pour leur support et pour leur amour inconditionnel. J'aimerais aussi remercier mes frères et sœurs pour leur confiance et encouragement tout au long de mes études. Enfin, ma jolie femme, qui a la lourde tâche de me supporter, quelle que soit mon humeur, merci d'être à mes côtés et de me soutenir.

Table des matières

Résumé	iii
Remerciements	v
Table des matières	vi
Liste des abréviations	xv
Liste des symboles	xviii
Liste des tableaux	xxi
Liste des figures	xxiii
Chapitre 1 Introduction générale.....	1
1.1 Introduction.....	1
1.2 Context.....	2
1.3 Problématique	4
1.4 Objectif principal	4
1.5 Solutions proposées	5
1.5.1 Synthèse de scénarios, stratégie d'échantillonnage et simulations.....	5
1.5.2 Évaluation et classification des risques	6
1.5.3 Validation	7
1.6 Contributions et originalités du projet de recherche	9
1.7 Liste des publications et rapports.....	11
1.7.1 Liste des publications	11
1.7.2 Liste des rapports.....	12
1.8 Structure de la thèse	12

Chapitre 2 État de l’art	14
2.1 Généralités	14
2.1.1 Les véhicules intelligents.....	14
2.1.2 Les véhicules autonomes	16
2.1.3 Les véhicules connectés.....	19
2.1.4 Évaluation des risques	20
2.2 Approches de test et validation des véhicules intelligents/ autonomes	21
2.3 Évaluation sur pistes d’essai	21
2.3.1 Documents pertinents actuels et au niveau mondial relatif à la normalisation des véhicules intelligents/connectés/autonomes	24
2.4 Évaluation par simulation	24
2.4.1 Pendant la phase de développement	24
2.4.2 Plateformes de simulation véhiculaire.....	26
2.4.3 Approches probabilistes	28
2.5 Évaluation par essais sur route (tests opérationnels sur terrain)	34
2.5.1 Projets FOT des systèmes avancés d’aide à la conduite (ADAS)	35
2.5.2 Projets FOT des systèmes de communication véhiculaire (V2X).....	36
2.6 Méthodes d’apprentissage automatique et ensembliste	38
2.7 Conclusion	39
Chapitre 3 Nouvelle stratégie d’échantillonnage de tests opérationnels sur terrain (FOT) basée sur l’algorithme Metropolis-Hastings	40
3.1 Introduction.....	40
3.2 Description du système ADAS étudié	41
3.2.1 Principe de fonctionnement d'un système AEB	42

3.2.2	Fonctionnalités et cas d'utilisations	42
3.3	Traitement de la base de données FOT.....	43
3.3.1	Données DAS1	45
3.3.2	Données DAS2	46
3.3.3	Données météorologiques	46
3.4	Estimation des modèles de fonctions de densités de probabilités (PDF)	47
3.5	Implémentation de l'algorithme : Metropolis-Hastings.....	48
3.6	Choix de l'écart type optimal (σ).....	49
3.7	Résumé des résultats de la stratégie d'échantillonnage proposée.....	53
3.8	Conclusion	54
Chapitre 4 Modified Metropolis-Hastings Algorithm for Efficient Sampling from Discrete Probability Distributions (MMH-DPD).....		55
4.1	Avant-propos	55
4.1.1	Auteurs et affiliation.....	55
4.1.2	Date de soumission.....	55
4.1.3	État de l'acceptation	55
4.1.4	Revue.....	Error! Bookmark not defined.
4.1.5	Référence	56
4.1.6	Titre français.....	56
4.1.7	Résumé français.....	56
4.2	Abstract	57
4.3	Introduction.....	57
4.4	Original Metropolis-Hastings Algorithm.....	59
4.4.1	Example: Relative Velocity distribution	60

4.5	Modified Metropolis-Hastings algorithm for discrete probability distributions	62
4.5.1	Using multimodal distribution.....	63
4.5.2	Using mixture of Generalized Gaussians	65
4.6	Generalization: Modified Metropolis-Hastings Algorithm for Discrete Probability Distributions (MMH-DPD)	69
4.7	Conclusion	73
 Chapitre 5 Proposal of a New Virtual Evaluation Approach of Preventive Safety Applications and Advanced Driver Assistance Functions - Application: AEB System -		
5.1	Avant-propos	74
5.1.1	Auteurs et affiliation.....	74
5.1.2	Date de soumission.....	74
5.1.3	État de l'acceptation	74
5.1.4	Revue.....	74
5.1.5	Référence.....	75
5.1.6	Titre français.....	75
5.1.7	Résumé français.....	75
5.2	Abstract.....	76
5.3	Introduction.....	76
5.4	Proposed evaluation approach	79
5.5	Scenarios synthesis	80
5.5.1	AEB system description	80
5.5.2	Accidents classification	80
5.5.3	Evaluation indicators	80

5.5.4	Most Representative Factors of AEB Scenarios.....	82
5.5.5	Proposed test scenarios (synthesis)	82
5.6	Sampling strategy and simulations	83
5.6.1	SPMD database	83
5.6.2	Meteorological data	84
5.6.3	Data Extraction	84
5.6.4	Probability density function (PDF) estimation	85
5.7	Metropolis-Hastings Algorithm Implementation.....	87
5.8	Scenario implementation (PreScan Simulator).....	89
5.9	Simulation Steps and Results.....	89
5.10	Risk assessment and classification.....	90
5.10.1	Risk assessment	90
5.10.2	Risk classification.....	92
5.11	Conclusion.....	94
5.12	Appendix	95
5.12.1	PDF model's estimation results	95
5.12.2	Metropolis-Hastings sampling results	98
Chapitre 6 Worst-case Scenarios Identification Approach for the Evaluation of Advanced Driver Assistance Systems in Intelligent/Autonomous Vehicles under Multiple Conditions.....		100
6.1	Avant-propos	100
6.1.1	Auteurs et affiliation.....	100
6.1.2	Date de soumission.....	100
6.1.3	État de l'acceptation	100

6.1.4	Revue.....	101
6.1.5	Titre français.....	101
6.1.6	Résumé français.....	101
6.2	Abstract.....	102
6.3	Introduction.....	102
6.4	Proposed validation approach.....	105
6.5	Predictive model of the studied AEB system	107
6.5.1	Physical track tests	107
6.5.2	Predictive model construction	109
6.6	Sampling the FOT Database	120
6.6.1	Multivariate Metropolis-Hastings algorithm (Component-Wise Updating)	121
6.6.2	Sampling results	123
6.7	Prediction step.....	123
6.7.1	Prediction results	125
6.8	Risk assessment	125
6.8.1	Risk assessment results.....	126
6.9	Risk classification	127
6.9.1	Case Study 1: 10000 Samples	127
6.9.2	Case Study 2: 250000 Samples	130
6.10	Final Selection [Worst-Case Scenarios].....	132
6.10.1	Zone L16: Notes < 0.15 & Speed ≥ 95	133
6.11	Web applications	135
6.11.1	Web application 1	135
6.11.2	Web application 2.....	137

6.12	Conclusion.....	137
6.13	Appendix	140
6.13.1	Results: Sampling the FOT database.....	140
6.13.2	Results: Risk classification.....	141
6.13.3	Results: Final selection [Worst-case scenarios]	141
Chapitre 7 Conclusion générale		143
7.1	Conclusions.....	143
7.2	Limitations et perspectives de recherche	146
7.2.1	Integration d’une solution HIL ou VeHIL ou ViL	146
7.2.2	Amélioration de la stratégie d’échantillonnage	147
7.2.3	Application des techniques de reverse engineering pour l’identification des systèmes ADAS.....	148
7.2.4	Application de techniques avancées de classification et de clustering.....	148
Annexe A Méthodes d’apprentissage automatique et ensembliste pour l’évaluation des véhicules intelligents/autonomes		149
A.1	Introduction.....	149
A.1.1	Méthodes de régression.....	149
A.1.1.1	Régression linéaire simple.....	149
A.1.1.2	Régression linéaire multiple (MLR).....	151
A.1.1.3	Régression à vecteur de support (SVR).....	152
A.1.1.4	Régression par processus gaussien (GPR)	155
A.1.2	Méthodes de classification	156
A.1.2.1	Régression logistique (LR).....	156
A.1.2.2	Classification naïve bayésienne (NB).....	157

A.1.2.3	Analyse discriminante linéaire et quadratique (LDA, QDA)	158
A.1.2.4	K-plus proches voisins (K-NN)	159
Annexe B Documents pertinents actuels et au niveau mondial relatif à la normalisation des véhicules intelligents/connectés/autonomes		163
B.1	National Highway Traffic Safety Administration (NHTSA)	163
B.1.1	Safety Technologies	163
B.1.2	Vehicle to Vehicle Communications.....	163
B.1.3	Vehicle Cybersecurity	163
B.1.4	Automated Vehicles	164
B.2	European New Car Assessment Program (Euro NCAP)	164
B.2.1	Safety Assist	164
B.2.2	Vulnerable Road User (VRU) Protection	164
B.3	International Organization For Standardization (ISO)	164
B.3.1	Advanced Driver Assistance Systems.....	164
B.3.2	Vehicle To X Communications.....	165
B.3.3	Automated Vehicles	165
B.4	Society of Automotive Engineers (SAE).....	166
B.4.1	Active Safety Systems Standards Committee	166
B.4.2	Safety and Human Factors standards Steering Committee	166
B.4.3	DSRC Tech Committee (Dedicated Short Range Communication)	167
B.4.4	On-Road Automated Driving (ORAD) Committee	167
Annexe C Proposition d’une plateforme de test et validation MIL/SIL/PIL/HIL pour PMG Technologies		169
C.1	Objectifs de la plateforme	169

C.2	Principales caractéristiques de la plateforme	169
C.3	Logiciels nécessaires.....	169
C.4	Matériels nécessaires	169
C.5	Représentation visuelle de la plateforme	170
C.6	Schéma de fonctionnement la plateforme.....	171
C.7	Clients potentiels.....	171
C.8	Estimation budgétaire	171
Références bibliographiques		172
LISTE DES PUBLICATIONS		190
LISTE DES RAPPORTS		191

Liste des abréviations

ABS	Anti-lock Braking System
ACC	Adaptive Cruise Control
AdaBoost	Adaptive Boosting
ADAC	Allgemeiner Deutscher Automobil Club
ADAS	Advanced Driver Assistance Systems
ADS	Automated Driving System
AEB	Autonomous Emergency Braking
AUC	Area Under The Curve
BSM	Basic Safety Message
BSMS	Blind Spot Monitoring System
CMVSS	Canada Motor Vehicle Safety Standards
CYMX	Aéroport International Montréal-Mirabel
DARPA	Defense Advanced Research Projects Agency
DSRC	Dedicated Short Range Communications
ECU	Electronic Control Unit
ESP	Electronic Stability Program
EURONCAP	European New Car Assessment Program
FCW	Forward Collision Warning
FESTA	Field Operational Test Support Action
FMVSS	Federal Motor Vehicle Safety Standards
FOT	Field Operational Tests
FPGA	Field-Programmable Gate Array
GMM	Gaussian Mixture Model
GPS	Global Positioning System
HIL	Hardware In the Loop
IAV	Intelligent and Autonomous Vehicles
IEEE	Institute of Electrical and Electronics Engineers

IS	Importance Sampling
ISO	International Organization for Standardization
KDE	Kernel Density Estimation
LDW	Lane Departure Warning
Lidar/Ladar	Light/Laser Detection And Ranging
LTE	Long-Term Evolution (Telecommunication)
MADIS	Metorological Assimilation Data Ingest System
Matlab	Matrix Laboratory
MBT	Model-Based Development
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
MDD	Model Driven Development
MGG	Mixture of Generalized Gaussians
M-H	Metropolis-Hastings
MIL	Model In the Loop
MMH-MPD	Modified Metropolis-Hastings Algorithm for Efficient Sampling from Discrete Probability Distributions
NCAP	New Car Assessment Program
NHTSA	National Highway Traffic Safety Administration
OBU	On-Bord Unit
OTS	Office of Traffic Safety
PDF	Probability Density Function
PIL	Processor In the Loop
Pro-SiVIC	Simulateur Véhicule-Infrastructure-Capteur
PSW	Personal Weather Station
QMIC	Qatar Mobility Innovations Center
RADAR	Radio Detection And Ranging
RCC	Regulatory Cooperation Council
RMSE	Root Mean Squared Error
ROC	Receiver Operation Characteristics
RSE	Road Side Equipment

RSU	Road Side Unit
RT-LAB	Real Time LABoratory
SAE	Society for Automobile Engineers
SAIL	Stanford Artificial Intelligence Laboratory
SIA	Société des Ingénieurs de l'Automobile
SIL	Software In the Loop
SiVIC	Simulateur Véhicule-Infrastructure-Capteur
SPMD	Safety Pilot Model Deploiment
SRR	Short Range Radar
SSE	Sum of Squared Errors
SSV	Strikeable Surrogate Vehicle
STATS 19	Road Safety Data Statistics: Accident, Casualties and Vehicle Tables
TC	Transport Canada
TJA	Traffic Jam Assistant
TTC	Time To Collision
USDOT	US Department of Transportation
USNCAP	United States New Car Assessment Program
V2I	Vehicle To Infrastructure
V2V	Vehicle To Vehicle
V2X	Vehicle To X (Vehicle, Infrastructure, Pedestrian, etc.)
VA	Véhicule Autonome
VANET	Vehicular Ad-hoc Network
VEHIL	Vehicle Hardware in the Loop
VIA	Véhicule Intlligent et Autonomes
WAVE	Wireless Access in Vehicular Environment
XIL	X (Model, Processor, Software, Hardware) In the Loop

Liste des symboles

(par ordre d'apparition)

ρ	Niveau de performance
p	Probabilité
$\hat{p}_N(\gamma)$	Probabilité empirique
q_1, q_2, \dots	N échantillons indépendants identiquement distribués (iid)
\mathcal{Q}	Espace de parameres q_1, q_2, \dots
$f_{\mathcal{Q}}$	Fonction de densité de probabilité de \mathcal{Q}
J	Fonction indicatrice
σ^2	Variance
E	Espérance mathématique
ε	Précision de l'estimation
$1 - \delta$	Degré de confiance
N_{ch}	Nombre d'expériences calculé par l'application de l'inégalité de Chernoff
$\xi_{\mathcal{Q}}$	PDF strictement positive sur \mathcal{Q}
φ_1, \dots	N_{IS} échantillons iid (Importance Sampling)
$T(x' x)$	Matrice de transition
$p^{(0)}$	Distribution initiale de x
$p^{(t)}$	Distribution à l'itération t
π	Distribution stationnaire d'une chaîne de Markov

fp	Probabilité de premier passage
$r(x)$	Temps de retour pour l'état x
$F(x)$	Fonction de distribution cumulative de la distribution cible
$G(x)$	Fonction de distribution cumulative de la distribution de l'échantillon
$d_U(F, G)$	Test de Kolmogorov-Smirnov
σ	Écart type
h	Décision du test Kolmogorov-Smirnov
p -value	Valeur asymétrique du test Kolmogorov-Smirnov
k	Statistique du test Kolmogorov-Smirnov
$\theta^{(t)}$	État de la chaîne de Markov à l'itération t
$p(\theta)$	Distribution cible continue
α	Probabilité d'acceptation de la proposition
$N(\theta^{(t-1)} \sigma^2)$	Distribution normale centrée à l'état actuel
$gg(x; \mu, \sigma, \rho)$	Fonction de densité de probabilité d'une gaussienne généralisée
$\Gamma(\cdot)$	Fonction Gamma
$p_M(x)$	Densité de probabilité d'un mélange de gaussiennes généralisées
$\hat{f}_h(x)$	Estimateur non paramétrique par la méthode d'estimation par noyau
$K(x)$	Distribution normale standard ($\mu = 0$ et $\sigma = 1$)
$g_j(x)$	Distribution normale multidimensionnelle pour le cluster j
Σ_j	Matrice de covariance
$ \Sigma_j $	Determinant de la matrice de covariance
Σ_j^{-1}	Inverse de la matrice de covariance
w_j^i	Probabilité qu'un point i fait partie du cluster j

Φ_j	Probabilité à priori du cluster j
$T_{\max sn}$	Température la plus élevée enregistrée pendant un jour de neige au cours des 10 dernières années à Montréal
$point_{test}$	Nombre de points disponible pour les différentes vitesses
C_j	Classificateur j
R^2	Coefficient of determination
y_i	Valeur de la mesure i
\hat{y}_i	Valeur prédite correspondante
\bar{y}	Moyenne des mesures
$d(x, y)^2$	Carré de la distance euclidienne

Liste des tableaux

Tableau 2-1 Niveaux d'automatisations des VAs [41].....	17
Tableau 2-2 Exemple d'une matrice de tests pour un système AEB.....	23
Tableau 2-3 Méthodes d'apprentissages automatiques et ensemblistes utilisées.....	38
Tableau 3-1 Méthodes MC en un coup d'œil.....	48
Tableau 3-2 Méthodes MCMC en un coup d'œil	49
Tableau 3-3 Tableau récapitulatif: écart type optimal (σ).....	53
Table 4-1 Discrete probability distribution.....	62
Table 4-2 Summary table: Original Distribution vs New Distribution (Multimodal distribution)	65
Table 4-3 Summary table: Original Distribution vs New Distribution (MGG)	69
Table 5-1 Scenarios synthesis.....	83
Table 5-2 Examples of samples provided to simulator	89
Table 5-3 Examples of data recovered and calculated after each simulation.....	90
Table 5-4 Scoring results of a few tests.....	92
Table 6-1 Test matrix	109
Table 6-2 Example: encoding categorical data.....	112
Table 6-3 Used machine learning techniques.....	113
Table 6-4 Prediction results of the first 6 outputs	117
Table 6-5 Classification results of Impact (yes/no) output.....	118
Table 6-6 Prediction results of the last 2 outputs	119
Tableau 6-7 Examples of samples provided to the prediction model.....	124
Tableau 6-8 Descriptive statistics of the data set [10000 samples].....	124

Tableau 6-9 Examples of data collected and calculated after each prediction	125
Tableau 6-10 Examples of some scoring results	126
Tableau 6-11 Descriptive statistics [Zone L16]	134
Tableau 6-12 Inputs – Predicted outputs – Real outputs [web application 1]	135
Tableau 7-1 Exemple de classification [K-NN]	161

Liste des figures

Figure 1-1 Approche d'évaluation proposée « Évaluation Combinée » [Version Définitive]	5
Figure 1-2 Synthèse de scénarios, stratégie d'échantillonnage et simulations	6
Figure 1-3 Évaluation et classification des risques	7
Figure 1-4 Validation	8
Figure 2-1 Boucle de développement d'un système de véhicule intelligent	15
Figure 2-2 Architecture de base d'un VA [35]	18
Figure 2-3 Applications des différents capteurs dans un VA [35]	18
Figure 2-4 Communications V2X (Source : Qatar Mobility Innovations Center (QMIC))	20
Figure 2-5 Détermination des niveaux ASIL : évaluation des risques	21
Figure 2-6 Principe de fonctionnement du VEHIL de TNO Automobile [10]	26
Figure 3-1 Architecture d'évaluation proposée	44
Figure 3-2 Contenu de la base de données SPMD [162]	45
Figure 3-3 Scénario de test étudié	47
Figure 3-4 Taux d'acceptation en fonction de l'écart type (σ) [Vitesse relative]	50
Figure 3-5 Résultats d'échantillonnage ($\sigma = 2$) : Vitesse Relative [2000 itérations]	51
Figure 3-6 Résultats d'échantillonnage ($\sigma = 6$) : Vitesse Relative [2000 itérations]	52
Figure 3-7 Résultats d'échantillonnage ($\sigma = 200$) : Vitesse Relative [2000 itérations]	52
Figure 4-1 Relative velocity probability density	61
Figure 4-2 Sampling results ($\sigma = 6$): Relative Velocity [2000 iterations]	61
Figure 4-3 Discrete probability distribution: (a) events (b) events replaced with integers	62
Figure 4-4 Estimated continuous probability density function of meteorological events	63
Figure 4-5 Sampling results of the continuous distribution	64

Figure 4-6 New probability distribution of the retrieved samples by the modified sampler based on a multimodal distribution	64
Figure 4-7 Generalized Gaussians with different locations, variance and shape parameters (μ, σ, ρ).....	66
Figure 4-8 Continuous probability density distribution of discrete events (meteorological events) using a mixture of Generalized Gaussians (MGG).....	67
Figure 4-9 Sampling results of the continuous distribution (MGG)	68
Figure 4-10 New probability distribution of the retrieved samples by the modified sampler based on a mixture of Generalized Gaussians	68
Figure 5-1 Adapted V-shaped design and evaluation cycle (PREVAL project) [171]	77
Figure 5-2 Combination of the three validation approaches	78
Figure 5-3 Proposed evaluation approach	79
Figure 5-4 Types of accidents (a) Vehicle with vehicle, (b) Vehicle with pedestrian	81
Figure 5-5 Most representative factors of AEB scenarios.....	82
Figure 5-6 PreScan GUI of the first scenario AEBJ1-1	84
Figure 5-7 Markov Chain diagram	88
Figure 5-8 Scoring table	91
Figure 5-9 Scoring histogram (score = 1–Note)	93
Figure 5-10 1/Distance as a function of scores (risk levels) (score = 1–Note).....	93
Figure 5-11 Risk classification: 12 levels (score = 1–Note).....	94
Figure 5-12 PDF estimation: Relative Distance D_{st}	95
Figure 5-13 PDF estimation: Relative Velocity V_s	96
Figure 5-14 PDF estimation: Temperature.....	96
Figure 5-15 PDF estimation: Humidity	97
Figure 5-16 PDF estimation: Visibility	97

Figure 5-17 Sampling results ($\sigma = 6$): Relative Velocity (2000 iterations).....	98
Figure 5-18 Sampling results ($\sigma = 4$): Relative Distance (2000 iterations).....	98
Figure 5-19 Sampling results ($\sigma = 5$): Temperature (2000 iterations)	98
Figure 5-20 Sampling results ($\sigma = 6$): Humidity (2000 iterations).....	99
Figure 5-21 Sampling results ($\sigma = 4$): Visibility (2000 iterations).....	99
Figure 6-1 Proposed evaluation approach	105
Figure 6-2 Proposed validation approach	106
Figure 6-3 (a) Followed approach -(b) Inputs/outputs prediction model - (c) Summary diagram	110
Figure 6-4 Ensemble Learning: Majority Voting Classifier [64]	114
Figure 6-5 (a) ROC Curve of each classifier - Learning Curves: (b) Speed at impact - (c) Shortest Distance	119
Figure 6-6 Sampled vs Actual bivariate distribution [Speed and Headway]	122
Figure 6-7 Sampling results: (a) Test Scenario (10000 samples) - (b) Target speed (10000 samples).....	123
Figure 6-8 Scoring table	126
Figure 6-9 (a) Headway = f (Notes) - (b) Speed= f (Notes) - (c) Deceleration = f (Notes) - (d) Target speed = f (Notes) : [13% of 10000 samples].....	128
Figure 6-10 K-Means clustering results - 4 clusters - 10000 samples: (a) Headway - (b) Speed - (c) Elbow Method - Optimal Number of Clusters = 4.....	129
Figure 6-11(a) Headway = f (Notes) - (b) Speed= f (Notes) - (c) K-Means clustering results: Headway – and (d) Speed [250000 samples - 4 clusters].....	131
Figure 6-12 Risk classification: 16 risk zones.....	132
Figure 6-13 (a) ZoneL16 : Notes < 0.15 & Speed \geq 95 - (b) Bar graph of each input parameter	133

Figure 6-14 Web application 1 : (a) Interface – (b) Prediction results - Web application 2: (c) Interface – (d) Upload excel file and data description sections – (e) Exploratory data analysis - (f) Prediction results – (g) Exploratory prediction analysis – (h) Risk classification	136
Figure 6-15 Sampling results - 10000 samples: (a) Deceleration - (b) Target - (c) Ambient temperature - (d) Wind speed - (e) Wind direction - (f) Offset - (g) Yaw rate	140
Figure 6-16 K-Means clustering - 10000 samples: (a) Distance - 6 clusters - (b) Distance - 12 clusters - (c) Deceleration - 4 clusters - (d) Target speed - 4 clusters	141
Figure 6-17 (a) Zone L15 - (b) Zone L14 - (c) Zone L13 - (d) Zone L12 - (e) Zone L11 - (f) Zone L10 - (g) Zone L9 - (h) Zone L8	142
Figure 6-18 (a) Zone L7 - (b) Zone L6 - (c) Zone L5 - (d) Zone L4 - (e) Zone L3 - (f) Zone L2 - (g) Zone L1	142
Figure 7-1 Stades de développement, de démonstration et de déploiement.....	146
Figure 7-2 Test des applications ADAS avec la solution ViL [65].....	147
Figure 7-3 SVR linéaire à une dimension [r4]	153
Figure 7-4 Exemple de classification [K-NN].....	160

Chapitre 1

Introduction générale

1.1 Introduction

Selon les acteurs du monde automobile, les moyens de mobilités du futur seront autonomes, économes et connectés en particulier pour les transports routiers. Une robotisation de la conduite automobile, issue de différentes recherches et développement sur la mobilité autonome, aura deux impacts majeurs : l'élimination des erreurs de conduite d'origine humaine et de leurs conséquences parfois dramatiques, et l'accès à l'automobile pour de nouvelles catégories d'utilisateurs, comme les personnes âgées ou handicapées [1]. Pour la catégorie d'utilisateurs traditionnels, une mobilité autonome apporte un confort supplémentaire avec la possibilité de communiquer, de se divertir et de se détendre plutôt que de se concentrer sur leur trajet.

Une mobilité plus économe en énergie et en matières premières aura un impact réellement significatif sur l'efficacité de la circulation, sur le plan environnemental, créatrice de lien social et de bien-être et économiquement viable. Une mobilité connectée intègre des technologies électroniques et informatiques sans-fil afin de partager et d'accéder aux médias numériques et à des informations numériques, les modifier ou de les communiquer permettant une meilleure gestion et sécurité des déplacements routiers quotidiens. Par exemple, il est estimé dans le cadre du programme d'évaluation des STI [2] que les systèmes d'alerte de sortie de voie ont réduit tous les accidents pertinents de 11% et tous les accidents liés à des blessures de 21%.

Le développement des systèmes de transport intelligents est fondé principalement sur l'exploitation des technologies de l'information et de la communication, la collecte, le traitement et la diffusion, dans un système intégré, d'informations échangées automatiquement entre les infrastructures de transport et les véhicules dotés des dispositifs de télécommunication appropriés. Grâce à cette intégration de technologies, les usagers de la route seront de plus en plus en mesure d'utiliser des réseaux de transport efficaces, sécuritaires et durables [3].

1.2 Context

Les véhicules intelligents et autonomes (VIA) atteignent une complexité sans précédent, ce qui rend leur évaluation et leur validation encore plus complexes. Les systèmes avancés d'assistance au conducteur (ADAS) constituent la base du développement des véhicules autonomes [4] et passent par un processus de validation en plusieurs stades: développement, démonstration et déploiement [5]. Pour compléter les essais sur route, plusieurs chercheurs ont suggéré de mettre au point d'autres méthodes d'essais afin de faire progresser l'intégration et l'utilisation quotidienne des véhicules autonomes. Ces méthodes alternatives peuvent inclure la modélisation mathématique, les tests virtuels, les simulateurs véhiculaires, les tests par scénarios, les tests accélérés et les études pilotes [5].

Au cours de la phase de développement, plusieurs approches sont utilisées [6]: modèle dans la boucle (MIL : Model in the Loop), logiciel dans la boucle (SIL : Software in the Loop), processeur dans la boucle (PIL : Processor in the Loop) et matériel dans la boucle (HIL : Hardware in the Loop). Selon Pawel et al [7], le développement basé sur modèle (MDD : Model Driven Development) et les tests basés sur modèle (MBT : Model-Based Testing) sont les approches privilégiées pour le développement de systèmes ADAS. Une approche qui nécessite généralement une compréhension et une connaissance approfondies du fonctionnement du système. Les auteurs dans [8] ont présenté différentes approches X-in-the-loop (XiL) et leur validation par rapport à des tests sur terrain. Ils fournissent également un état de l'art sur différentes méthodes XiL et de validation dans lesquelles plusieurs simulations et normes de validation sont répertoriées. Pour une méthodologie plus efficace dans le domaine des véhicules et en particulier dans la validation des ADAS et des systèmes de communication intervéhicule, un nouveau concept a été proposé: véhicule et matériel dans la boucle (VEHIL : Vehicle Hardware in the Loop) par TNO Automobile [9]–[11]. Ce dernier consiste à intégrer un véhicule dans la boucle de validation, mais celui-ci reste dans un environnement de laboratoire. Les experts s'entendent pour dire que la maîtrise des efforts de validation à l'avenir ne peut être réalisée qu'au moyen de beaucoup de simulations [8]. Les auteurs dans [12] ont souligné l'importance d'utiliser une stratégie de simulation dans le processus de développement de systèmes de sécurité active et d'assistance avancée au conducteur, dans lesquels ils expliquaient comment effectuer des simulations en temps réel et en temps non réel sur une machine à

processeurs multicœurs. La validation des ADAS doit aborder les aspects déterministes (concept de sécurité) et non déterministes en combinant des tests de conduite réels et des simulations.

Les méthodes d'évaluation et de validation souvent utilisées sont beaucoup plus déterministes. Cela signifie que de nombreux facteurs, tels que la vitesse des véhicules, sont traités comme s'ils étaient fixes et connus avec précision. Ils tentent de prendre en compte la variabilité et l'incertitude à l'aide de facteurs de sécurité fixes. Par ailleurs, les approches probabilistes permettent de quantifier les variations et les incertitudes en utilisant principalement des distributions de probabilités plutôt que des valeurs fixes. Dans cette même approche, certaines initiatives ont vu le jour dans le domaine des véhicules intelligents et autonomes (VIA) [14]–[17].

Pour les stades de démonstration et de déploiement, deux approches principales sont utilisées: les pistes fermées et les essais sur route (ou tests opérationnels sur le terrain). Les essais en circuit fermé sont généralement effectués dans des centres d'essai spécialisés tels que le Centre d'essai et de recherche de véhicules automobiles de Transports Canada, géré par PMG Technologies [18]. Les tests opérationnels sur le terrain (en anglais FOT : Field Operational Tests) sont des programmes de test à grande échelle conçus pour fournir une évaluation plus complète de l'efficacité, de la qualité, de la robustesse et de l'acceptation par l'utilisateur des nouvelles technologies de véhicule telles que la navigation, les systèmes de communication, les systèmes avancés d'assistance au conducteur (ADAS) et les systèmes coopératifs. Plusieurs projets ont vu le jour au cours des vingt dernières années [19]–[23].

Le modèle de développement en V de la norme ISO 26262, largement adopté par l'industrie automobile, établit un cadre qui relie chaque essai automobile à un document de "spécification" correspondant. Dans [24], les auteurs ont abordé les défis à relever (conducteur en dehors de la boucle, complexité des exigences, algorithmes non déterministes, algorithmes d'apprentissage inductif et systèmes opérationnels défaillants) pour pouvoir utiliser ce modèle afin de garantir la sécurité d'un véhicule isolé donné. *"En général, le modèle du cycle en V représente un processus méthodique de création suivi d'une vérification et d'une validation [24]"*. Plusieurs procédures ont été proposées pour le développement des ADAS telles que : Code of Practice, des Standards ISO (ISO 26262) et les lignes directrices CONVERGE. *"Malheureusement, ils ne peuvent être appliqués qu'à un niveau d'abstraction élevé dans le processus de*

développement. Ils ne fournissent pas d'exigences objectives ni de critères d'évaluation pour la validation et l'étalonnage des ADAS et ne prescrivent pas l'utilisation d'outils et de méthodes spécifiques dans le processus de validation [16]." Dans [5], les auteurs ont présenté un cadre définissant comment mesurer la sécurité des véhicules autonomes (véhicules hautement automatisés > niveau 4), en trois stades: développement, démonstration et déploiement. Dans la phase de déploiement, ils incluent uniquement l'évaluation sur les routes publiques et excluent la simulation et les essais sur piste (dans des circuits fermés).

1.3 Problématique

Pour démontrer les performances attendues d'un système avancé d'aide à la conduite (ADAS) dans un véhicule intelligent ou hautement automatisé, les méthodes d'évaluation doivent inclure des simulations, des essais sur piste et des essais sur route (essais sur terrain/essais réels) [25]. Les VIA continueront à rendre la conduite plus facile et plus sécuritaire. Néanmoins, la question ultime reste à savoir quelle est l'approche d'évaluation optimale à proposer, qui aura la capacité de valider le comportement et les performances attendues liées au fonctionnement des systèmes embarqués dans des véhicules intelligents et autonomes pendant tous les stades; et avec des niveaux de confiance qui dépasse 95% : développement, démonstration et déploiement. À notre connaissance, la plupart des études/propositions qui tentent de combiner plusieurs approches (2 ou 3) sont appliquées pendant le stade de développement.

1.4 Objectif principal

L'objectif principal de notre projet de recherche consiste à proposer une nouvelle approche d'évaluation des systèmes avancés d'aide à la conduite dans un véhicule intelligent ou hautement automatisé conduit par une entité externe pendant la phase de déploiement (ou de rappel), qui a un accès limité aux informations du système d'aide à la conduite à tester (c.-à-d. sa composition matériel ou logiciel et/ou son modèle mathématique et/ou ses entrées/sorties) et lorsqu'il est traité comme boîte noire. Cette approche permet à la fois d'identifier un ensemble de pires scénarios pour une application ADAS donnée et de combiner les trois approches d'évaluation mentionnées précédemment (simulations, essais sur piste et essais sur route).

1.5 Solutions proposées

La version finale de l'approche d'évaluation proposée est divisée en trois parties (voir Figure 1-1): 1) Synthèse de scénarios, stratégie d'échantillonnage et simulations, 2) Évaluation et classification des risques, 3) Validation. La première et deuxième partie sont abordées au chapitre 5 [26]–[28]. La troisième partie est abordée plus en détail au chapitre 6 [29].

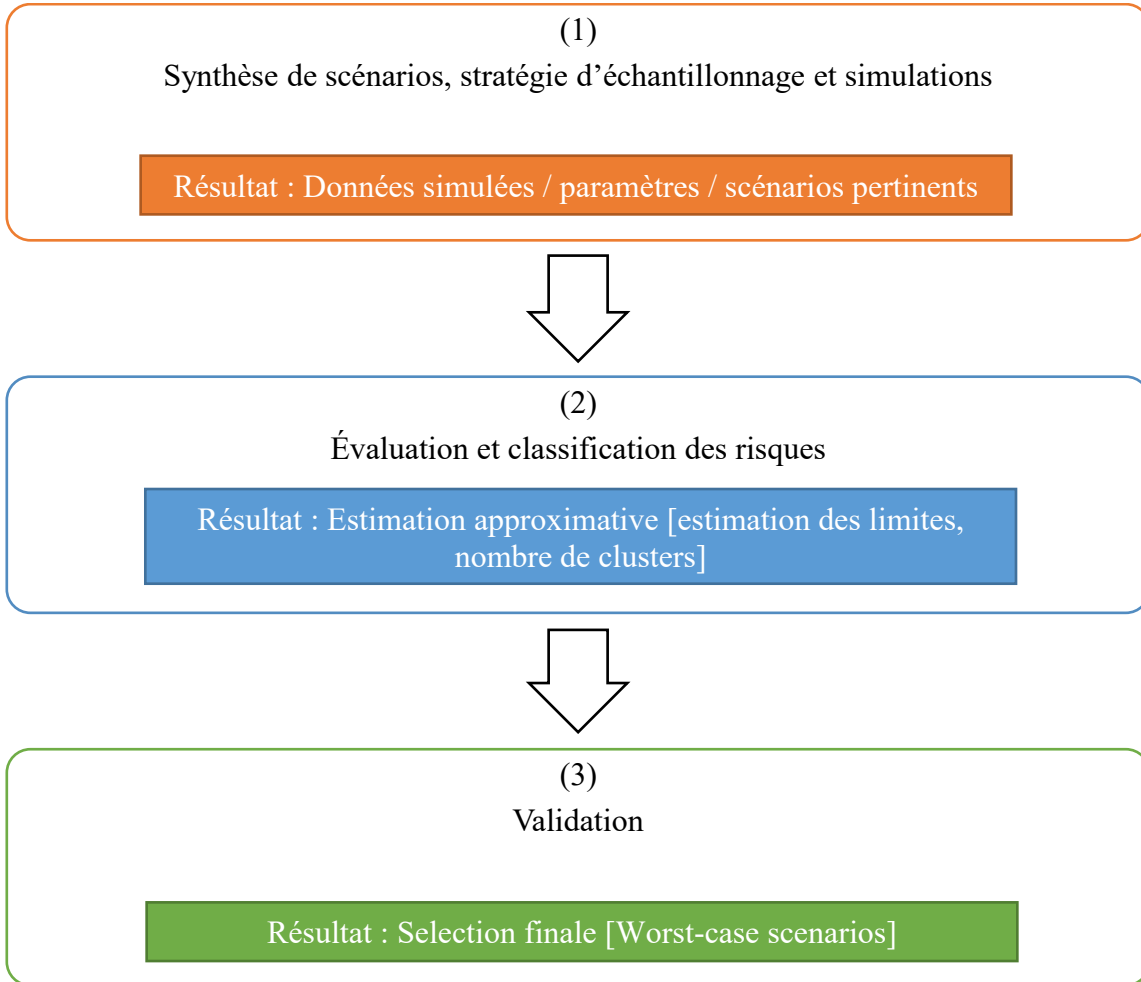


Figure 1-1 Approche d'évaluation proposée « Évaluation Combinée » [Version finale]

1.5.1 Synthèse de scénarios, stratégie d'échantillonnage et simulations

Cette première partie (voir Figure 1-2) comprend une description du système ADAS à évaluer et de ses différents critères d'évaluation, propose une synthèse des scénarios de test les plus pertinents avec les paramètres de fonctionnement de chaque scénario. Ensuite, nous traitons la

base de données FOT en implémentant une stratégie d'échantillonnage appropriée et à la fin de cette partie, des tests virtuels sont mis en oeuvre dans un environnement de simulation véhiculaire.

1.5.2 Évaluation et classification des risques

La deuxième partie (voir Figure 1-23) se concentre sur la collecte des résultats de simulation, puis sur l'évaluation et la classification du risque de chaque test, ce qui nous permet ensuite de récupérer les niveaux de risques et d'avoir une estimation approximative (estimation des limites, nombre de clusters, etc.) de l'ensemble de scénarios défavorables. Cette approximation sera ensuite considérée comme point de départ (initialisation) pour la partie validation.

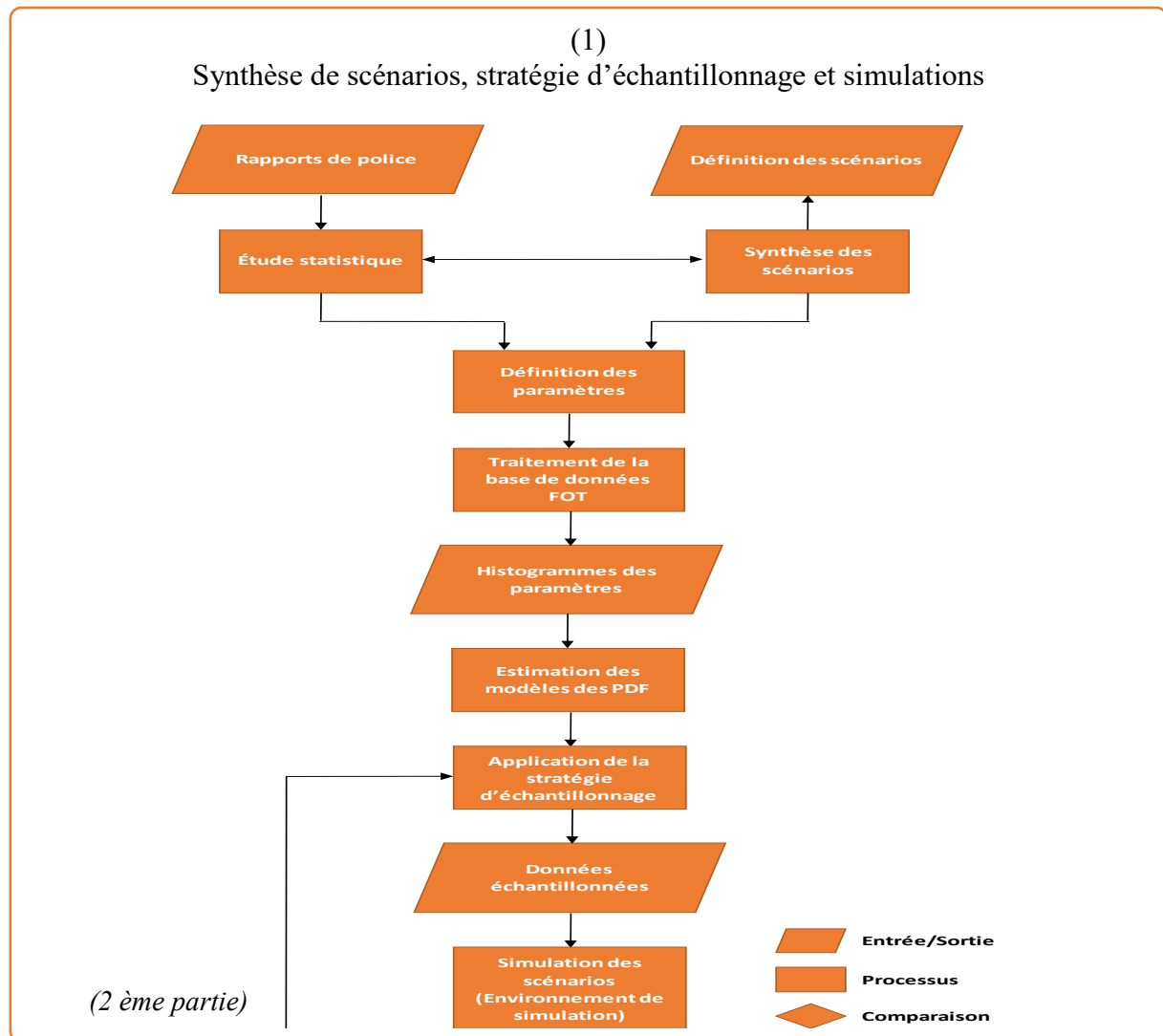


Figure 1-2 Synthèse de scénarios, stratégie d'échantillonnage et simulations

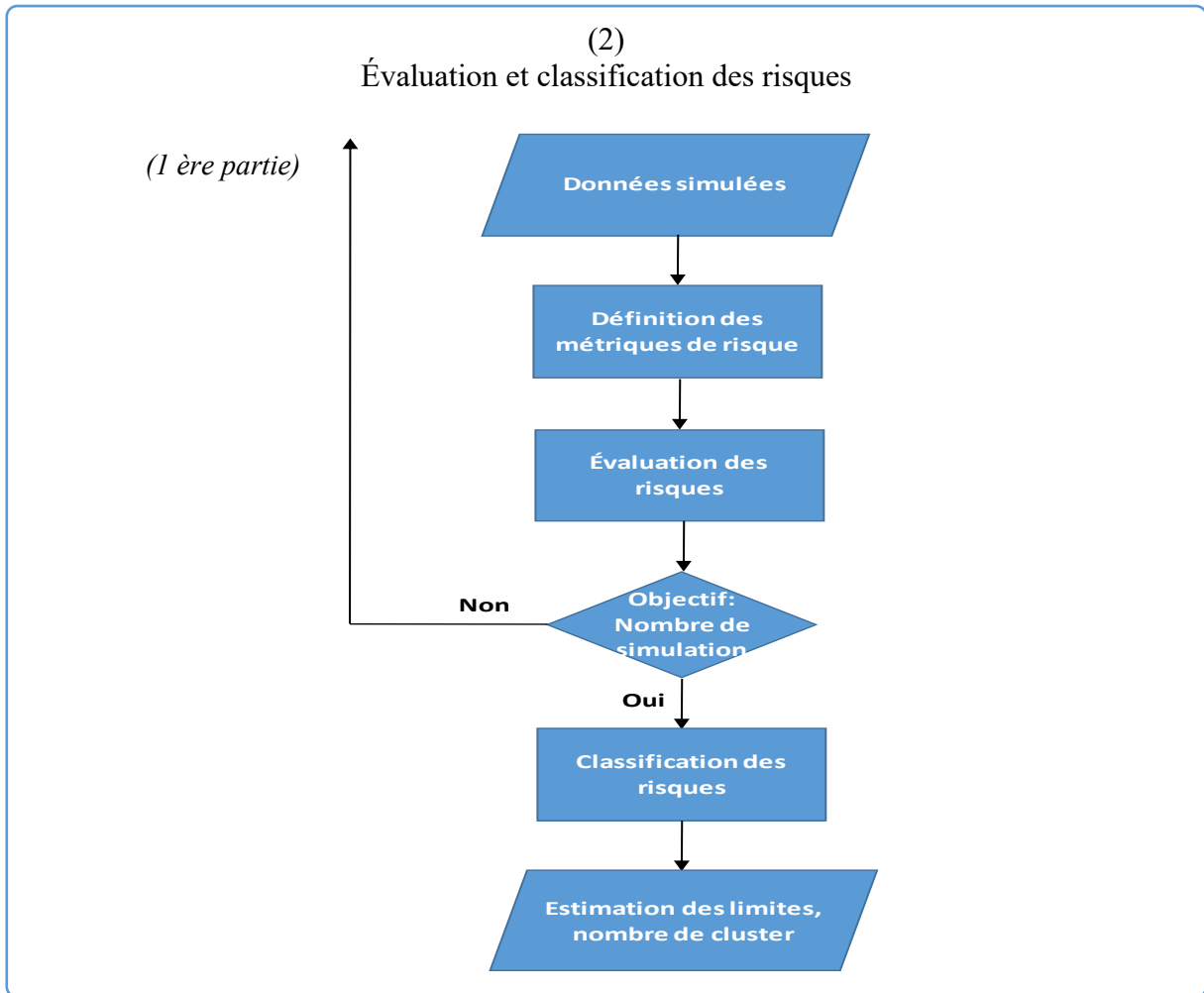


Figure 1-3 Évaluation et classification des risques

1.5.3 Validation

Nous avons divisé la partie validation en trois sous-parties (voir Figure 1-4). Une première partie (en noir), qui comprend une description des résultats des tests sur piste de l'ADAS étudié et des différentes techniques d'apprentissage automatique utilisées pour créer son modèle prédictif. Afin de diagnostiquer le surajustement (variance élevée) ou sous-ajustement (biais élevé) des résultats de prédictions, la notion de courbes d'apprentissages (Learning Curves) a été introduite. Avec cette technique, nous pouvons facilement détecter si le modèle présente une variance élevée ou un biais important, et si la collecte de plus de données pourrait aider à résoudre ce problème. Dans notre cas, elle nous permet de décider si un nombre supplémentaire d'essais physique est nécessaire ou non. Une seconde partie (en violet) qui traite encore une fois

la base de données FOT et l'implémentation d'une stratégie d'échantillonnage plus avancée. La troisième partie (en rouge) se concentre sur la collecte des résultats de prédiction, puis sur l'évaluation de risque de chaque test et sur sa classification à l'aide d'une technique de classification non supervisée, ce qui nous permet de construire et sélectionner finalement un ensemble des pires scénarios.

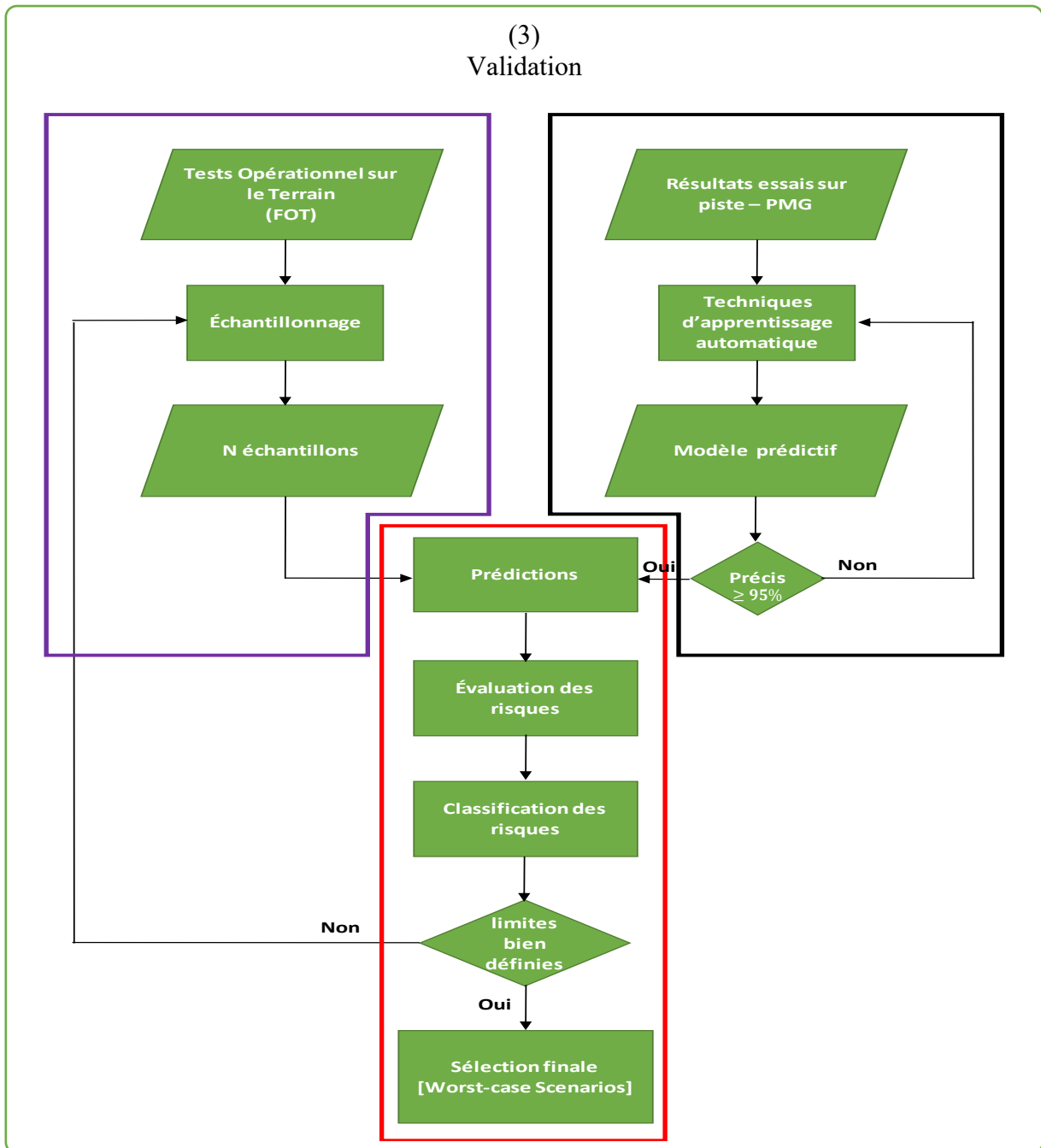


Figure 1-4 Validation

1.6 Contributions et originalités du projet de recherche

Nous proposons une nouvelle approche d'évaluation des systèmes avancés d'aide à la conduite dans un véhicule intelligent ou hautement automatisé – Approche d'« Évaluation Combinée ».

Les principales contributions de cette thèse peuvent être listées comme suit :

- Le concept « Évaluation Combinée », pendant la phase de déploiement, est nouveau dans le domaine des essais et d'évaluation de véhicules intelligents ou hautement automatisés. Il permet à la fois d'identifier un ensemble de scénarios défavorables et de combiner les trois approches d'évaluations, souvent utilisées séparément: simulations, essais sur piste et essais sur route. Cette approche est adaptée essentiellement dans un contexte où l'évaluation est conduite, pendant la phase de déploiement (ou de rappel), par une entité externe qui a un accès limité aux informations du système d'aide à la conduite à tester (c.-à-d. sa composition matériel et/ou logiciel et/ou son modèle mathématique et/ou ses entrées/sorties) et lorsqu'il est traité comme boîte noire.
- La proposition d'une nouvelle stratégie d'échantillonnage de tests opérationnels sur terrain (FOT) basé sur une combinaison entre l'algorithme Métropolis-Hastings (M-H) et le test de Kolmogorov-Smirnov. La stratégie d'échantillonnage est basée sur les méthodes de Markov Chain Monte-Carlo (MCMC), utilisant l'algorithme Metropolis-Hastings pour générer des échantillons à partir de distributions de probabilités de tests opérationnels sur terrain. Pour choisir l'écart type optimal (σ) pour la distribution de proposition, utilisé dans l'algorithme M-H, nous proposons de combiner les résultats du test de Kolmogorov-Smirnov avec le taux d'acceptation de chaque écart type.
- Pour un échantillonnage plus efficace à partir des distributions de probabilités discrètes, nous avons proposé une nouvelle version de l'algorithme de Metropolis-Hastings « MMH-DPD » (Modified Metropolis-Hastings Algorithm for Efficient Sampling from Discrete Probability Distributions). Contrairement à l'algorithme d'origine (M-H), la nouvelle version peut tirer des échantillons à partir de distributions de probabilités discrètes.

- La proposition d'une nouvelle approche d'évaluation virtuelle des applications de sécurité préventive et de fonctions avancées d'aide à la conduite. L'approche décrit une suite méthodologique de toutes les étapes à suivre. Commenant par une description du système ADAS étudié et une synthèse des scénarios de tests les plus pertinents. Puis, l'application de plusieurs méthodes d'estimation afin de modéliser la distribution de chaque paramètre de test extrait à partir de la base de données FOT. Ces distributions sont ensuite échantillonnées à l'aide de la stratégie d'échantillonnage et la version modifiée de l'algorithme mentionnées précédemment. Et finalement, les échantillons de chaque paramètre récupérés avec cet échantillonneur sont simulés avec un logiciel de simulation véhiculaire.
- La proposition d'une nouvelle approche d'identification des pires scénarios pour l'évaluation de systèmes avancés d'assistance à la conduite dans des véhicules intelligents/autonomes soumis à de multiples conditions. Dans la première partie, les tests physiques du système ADAS et les différentes techniques d'apprentissage automatique et ensembliste utilisées pour créer son modèle prédictif sont présentés. Ensuite, une deuxième partie qui traite la base de données de tests opérationnels sur le terrain, de la mise en œuvre d'une nouvelle stratégie d'échantillonnage (plus avancée que la première) basée sur l'algorithme Metropolis-Hastings original, modifié et multivarié. La troisième partie porte sur la collecte des résultats de prédiction, puis sur l'évaluation de risque de chaque test, pour les classer à l'aide d'une technique de classification non supervisée, ce qui nous permet de construire un ensemble de pires scénarios et faire une sélection finale.
- Pour améliorer la précision des résultats de prédictions, plusieurs techniques d'apprentissage automatique (18 techniques) ont été étudiées et ensuite utilisées pour produire un classificateur basé sur un vote majoritaire (voir section 6.5.2).
- L'introduction de la notion de courbes d'apprentissages (Learning Curves) afin de diagnostiquer le surajustement (variance élevée) ou sous-ajustement (biais élevé) des résultats de prédictions. En traçant les précisions de prédictions de l'ensemble d'apprentissages et l'ensemble de validation en fonction de la taille de l'ensemble

d'apprentissages, nous pouvons facilement détecter si le modèle présente une variance élevée ou un biais important, et si la collecte de plus de données pourrait aider à résoudre ce problème. Dans notre cas, elle nous permet de décider si un nombre supplémentaire d'essais physique est nécessaire ou non.

- Le développement et le déploiement en ligne de deux applications web (Full-stack web applications). La première permet la prédiction en temps réel des sorties du système ADAS étudié pour un scénario donné. La deuxième application est une version améliorée de la première et qui procure plus d'options et de fonctionnalités. L'utilisateur peut : envoyer au serveur web un fichier Excel qui contient le nombre de scénarios qu'il veut, avoir une description statistique du jeu de données qu'il a envoyé au serveur, faire une analyse des données, voir les résultats de prédictions et de classification de chaque test en temps réel.

1.7 Liste des publications et rapports

1.7.1 Liste des publications

- (1) N. E. Chelbi, D. Gingras, and C. Sauvageau, "New Field Operational Tests Sampling Strategy Based on Metropolis-Hastings Algorithm," in *Intelligent Systems and Applications. IntelliSys 2018. Advances in Intelligent Systems and Computing*, 2019, vol. 868. (Publié)
- (2) N. E. Chelbi, D. Gingras, and C. Sauvageau, "Proposal of a New Virtual Evaluation Approach of Preventive Safety Applications and Advanced Driver Assistance Functions Based on Metropolis Hastings Algorithm - Application: AEB System -," *Proceedings, ITS World Congress (ITSWC 2018)*, Copenhagen, Denmark, Nov. 2018. (Publié)
- (3) N. E. Chelbi, D. Gingras, and C. Sauvageau, "Modified Metropolis-Hastings Algorithm for Efficient Sampling from Discrete Probability Distributions (MMH-DPD) Applied to Field Operational Tests database (SPMD)," in *IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, 2018, pp. 125–131. (Publié)
- (4) N. E. Chelbi, D. Gingras, and C. Sauvageau, "Proposal of a new virtual evaluation approach of preventive safety applications and advanced driver assistance functions – application: AEB system," *IET Intelligent Transport Systems*, vol. 12, no. 9, pp. 1148–1156, 2018. (Publié)

(5) N. E. Chelbi, D. Gingras, and C. Sauvageau, “Worst-case Scenarios Identification Approach for the Evaluation of Advanced Driver Assistance Systems in Intelligent/Autonomous Vehicles under Multiple Conditions,” *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, p. 34. (Soumis)

1.7.2 Liste des rapports

(1) Les véhicules connectés - Préparé par : Nacer Eddine Chelbi - (Document interne - PMG Technologies).

(2) Centre d’essais et de recherche de la mobilité intelligente et de l’innovation (CERMII) - Préparé par : Nacer Eddine Chelbi et Ayet Bagane - (Document interne - PMG Technologies).

(3) Étude de l’état de l’art en homologation, évaluation et essais des véhicules intelligents (actuel et au niveau mondial) - Préparé par : Nacer Eddine Chelbi - (Document interne - PMG Technologies)

(4) Proposition des scénarios envisageables pour les essais sur piste des véhicules intelligents - Préparé par : Nacer Eddine Chelbi - (Document interne - PMG Technologies)

(5) Étude des plateformes matérielles pouvant être utilisées pour le développement d’un protocole d’essais étendu impliquant des essais physiques et virtuels - Préparé par : Nacer Eddine Chelbi - (Document interne – PMG Technologies)

(6) Proposition d’une plateforme de test et validation MIL/SIL/PIL/HIL - Préparé par : Nacer Eddine Chelbi - (Document interne - PMG Technologies)

1.8 Structure de la thèse

Le reste de cette thèse est organisé comme suit. Une revue de la littérature et les différents concepts reliés à ce projet de thèse sont présentés dans le chapitre deux. Au chapitre trois, une nouvelle stratégie d’échantillonnage de tests opérationnels sur terrain (FOT) basée sur l’algorithme Metropolis-Hastings est développée. Le chapitre 4 présente une version modifiée de l’algorithme de Metropolis-Hastings pour un échantillonnage efficace à partir de distributions de probabilités discrètes (MMH-DPD). Dans le chapitre 5, nous proposons une nouvelle approche d’évaluation virtuelle des applications de sécurité préventive et de fonctions avancées

d'aide à la conduite - application: système AEB. Le sixième chapitre présente une approche d'identification des pires scénarios pour l'évaluation de systèmes avancés d'assistance à la conduite dans des véhicules intelligents/ autonomes soumis à de multiples conditions. Enfin et dans le dernier chapitre, nous résumons notre travail de recherche par des conclusions, des limitations de l'approche d'évaluation proposée ainsi que des perspectives pour de futurs travaux de recherche.

Chapitre 2

État de l'art

2.1 Généralités

2.1.1 Les véhicules intelligents

Hier, le véhicule était considéré essentiellement une machine mécanique, mais aujourd'hui, il contient de plus en plus de dispositifs électroniques et informatiques destinés principalement à l'amélioration de son contrôle : contrôle du moteur (allumage électronique), d'aide à la conduite (ABS, ESP, ACC, etc.) ou à la navigation via le GPS. Actuellement, ces dispositifs intègrent des fonctions encore plus avancées et plus complexes, d'aide à la décision et à la conduite : par l'échange d'informations intervéhiculaires ou entre véhicules et infrastructures via des systèmes de télécommunication conçus spécialement pour le domaine véhiculaire. Un échange qui permettra l'amélioration du contrôle de position ou de trajectoire en fonction du trafic et de l'environnement [30]. Un véhicule est considéré intelligent s'il a des capacités de mesure qui lui permet la collection des informations sur son environnement, de les traiter et les analyser afin de prendre des décisions pour prédire ou anticiper des risques. Un véhicule intelligent a aussi des capacités d'action, de communication et d'apprentissage [31].

2.1.1.1 Les capteurs dans un véhicule intelligent

Le véhicule intelligent est le système le plus complexe mis à la disposition du grand public. De nos jours, les véhicules intelligents sont équipés de plus d'une centaine de systèmes électroniques miniaturisés. Ces systèmes peuvent être catégorisés en trois groupes pour former la boucle de développement d'un système de véhicule intelligent : les capteurs, les actionneurs et les systèmes de traitement de l'information. Une telle approche a été bien développée dans [32], la Figure 2-1 ci-dessous récapitule le concept.

Les différents capteurs dans un véhicule sont regroupés en trois catégories [33] : des capteurs de véhicules en général [Tels que : capteur d'angle de lacet (Gyroscope), accéléromètre, centrale

inertielle, odomètre, capteur d'angle du volant, compas, GPS, capteur de température, capteur de pression des pneus, capteur de pluie], des capteurs de perception [Tels que : capteur Radar, scanner laser (LIDAR), système de vision (caméra) et les capteurs ultrasoniques] et des capteurs virtuels [Tels que : les cartes numériques (Digital Maps) et les communications sans-fil]. La majorité des capteurs de la première catégorie sont déjà disponibles sur le marché automobile dans la majorité des véhicules commercialisés. Par contre, le pourcentage de la pénétration des capteurs de perception dans le marché, à l'exception des capteurs ultrasons, est très faible principalement à cause de leurs coûts.

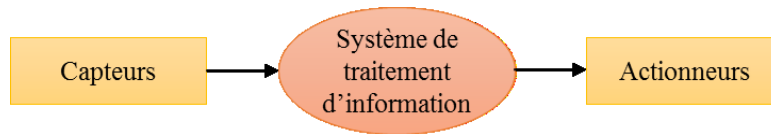


Figure 2-1 Boucle de développement d'un système de véhicule intelligent

2.1.1.2 Les systèmes d'aide à la conduite (ADAS)

La plupart des fonctions d'un véhicule intelligent/autonome sont contrôlées par des systèmes ADAS : « *ADAS (Advance Driving Assist System) est le terme anglais pour désigner les systèmes avancés d'assistance et d'aide à la conduite. Ces systèmes sont des systèmes électroniques ayant accès aux organes de restitution, motricité, freinage et direction du véhicule, permettant ainsi au conducteur de bénéficier d'une assistance et/ou de déléguer temporairement la conduite à un copilote automatique dans certaines conditions de circulation. Cet automate peut généralement être activé ou désactivé selon la volonté du conducteur. Les ADAS interviennent sur les différentes phases de conduite, avant la collision, ils participent à la sécurité primaire du véhicule (ou sécurité active)* » [34]. Le fonctionnement des ADAS est basé sur les différents capteurs décrits ci-dessus, ci-après une liste non exhaustive des différents ADAS soit existant sur le marché, soit sont en phase de lancement par certain constructeur :

- 1- Freinage d'urgence autonome/ Autonomous emergency braking (AEB) incluant l'Alerte de collision avant / Forward Collision Warning (FCW).
- 2- Système de surveillance d'angles morts / Blind spot monitoring system (BSMS).
- 3- Régulateur de vitesse adaptatif / Adaptive cruise contrôle (ACC).
- 4- Système d'aide au stationnement semi-automatique/ Park assist (PA).
- 5- Alerte de franchissement involontaire de ligne / Lane departure warning (LDW).

- 6- Système de suivi de voie / Lane keeping assist (LKA).
- 7- Limiteur de vitesse / Intelligent speed assistance (ISA).
- 8- Détecteur de fatigue et d'inattention / Driver drowsiness detection (DDD).
- 9- Systèmes de préparations en vue des accidents imminents/ Pre-crash.
- 10- Avis de circulation transversale / Cross-traffic alert (CTA).
- 11- Système d'identification de panneau de signalisation / Traffic sign recognition (TSR).
- 12- Système d'aide au trafic / Traffic jam assistant (TJA).
- 13- Système d'aide au freinage / Dynamic brake system/ (DBS).
- 14- Système de vision nocturne / Automotive night vision (ANV).
- 15- Pedestrian detection/Système de détection des piétons (PD).

2.1.2 Les véhicules autonomes

Depuis les années 1920, les gens ont toujours imaginé des véhicules sans conducteurs/autonomes (VAs) : « *Un véhicule pleinement autonome peut être défini comme étant un véhicule capable de percevoir et d'analyser son environnement, de décider quelle route prendre pour atteindre sa destination et d'exécuter par lui-même les manœuvres de conduites nécessaires. Un véhicule autonome signifie qu'il ne dépend plus de l'humain pour conduire. Sans la nécessité d'un conducteur, toute personne à bord est considérée comme un passager. Le véhicule peut même se conduire seul sans occupants* » [35]. Plusieurs prédictions prévoient une disponibilité limitée des fonctions de conduite automatisées d'ici 2020 (automatisation partielle et conditionnelle) et une large disponibilité d'ici 2040, incluant une automatisation élevée et complète [36]–[38].

En 2005 le véhicule robotisé « Stanley » a remporté le DARPA Grand Challenge, une compétition organisée par la DARPA (Defense Advanced Research Projects Agency aux États-Unis), mettant en jeu des véhicules terrestres sans pilote et autonomes. Ensuite, le projet « Google, Car » a vu le jour en 2010. Le développement des deux véhicules a été dirigé par le professeur Sebastien Thrun, Professeur à l'Université de Stanford et directeur du Stanford Artificial Intelligence Laboratory (SAIL), avec une équipe composée de Chris Urmson pour le développement logiciel, Anthony Levandowski pour le matériel et Mike Montemerlo pour la cartographie [39]. Après cet événement, plusieurs projets de recherches et de développements ont vu le jour (Navette Arma de NAVYA, Véhicule de Volvo, Uber et Lyft, Pilote automatique

de Tesla, etc. [40]) dans un seul objectif : comment rendre les véhicules plus intelligents et plus autonomes.

2.1.2.1 Niveaux d'automatisation des VAs

Les niveaux d'automatisation des VA, selon les définitions de SAE J3016 [41], sont résumés dans le Table 2-1. Ils s'étalent de l'absence d'automatisation à l'automatisation complète. Le tableau donne une définition des niveaux respectifs ainsi que des responsabilités et spécifie les exigences minimales. Le terme « system » désigne le système d'aide au conducteur ou le système de conduite automatisée.

Table 2-1 Niveaux d'automatisations des VAs [41]

Level	Name	Narrative definition	Sustained lateral and longitudinal vehicle motion control (steering and acceleration/deceleration)	Monitoring of driving environment	Fallback performance of dynamic driving task	System capability (driving modes)
Driver performs part or all of the dynamic driving task (DDT)						
0	No Driving Automation	The performance by the driver of the entire DDT, even when enhanced by active safety systems.	Driver	Driver	Driver	n/a
1	Driver Assistance	The sustained and operational design domain (ODD)-specific execution by a driving automation system of either the lateral or the longitudinal vehicle motion control subtask of the DDT (but not both simultaneously) with the expectation that the driver performs the	Driver and System	Driver	Driver	Some driving modes
2	Partial Driving Automation	The sustained and ODD-specific execution by a driving automation system of both the lateral and longitudinal vehicle motion control subtasks of the DDT with the expectation that the driver completes the object and event detection and response (OEDR) subtask and supervises the driving automation system.	System	Driver	Driver	Some driving modes
Automated driving system (ADS) ("System") performs the entire DDT (while engaged)						
3	Conditional Driving Automation	The sustained and ODD-specific performance by an ADS of the entire DDT with the expectation that the DDT fallback-ready user is receptive to ADS-issued requests to intervene, as well as to DDT performance-relevant system failures in other vehicle systems, and will respond appropriately.	System	System	Fallback-ready user (becomes the driver during fallback)	Some driving modes
4	High Driving Automation	The sustained and ODD-specific performance by an ADS of the entire DDT and DDT fallback without any expectation that a user will respond to a request to intervene.	System	System	System	Some driving modes
5	Full Driving Automation	The sustained and unconditional (i.e., not ODD-specific) performance by an ADS of the entire DDT and DDT fallback without any expectation that a user will respond to a request to intervene.	System	System	System	All driving modes

2.1.2.2 Architecture et complexité des VAs

La Figure 2-2 représente l'architecture de base d'un véhicule autonome. Les unités de commande des ADAS communiquent avec un environnement beaucoup plus complexe en utilisant des capteurs qui sont nouveaux dans l'industrie automobile. Une fusion de données est souvent utilisée pour combiner ces diverses sources d'informations et proposer une description améliorée et plus détaillée de son environnement extérieur.

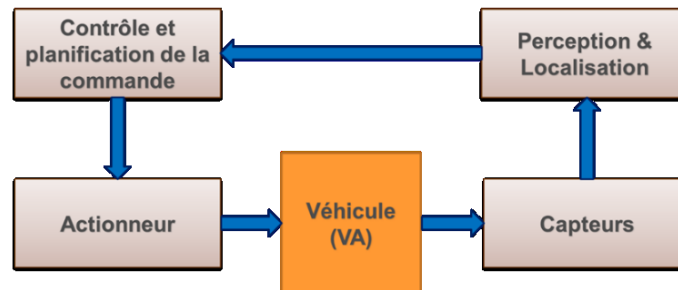


Figure 2-2 Architecture de base d'un VA [35]

Des informations sur le monde extérieur sont fournies par plusieurs capteurs dans des listes d'objets mises à jour périodiquement (capteurs vidéo, capteurs radar, capteurs LIDAR, capteurs ultrasons), voir Figure 2-3. D'autres informations sur l'environnement autour du véhicule sont également transmises via la communication véhicule à véhicule et la communication véhicule à infrastructure. Pour aboutir à une meilleure estimation du monde environnant le véhicule, toutes ces informations ainsi que les données classiques, telles que la vitesse du moteur, la vitesse, etc., sont utilisées.

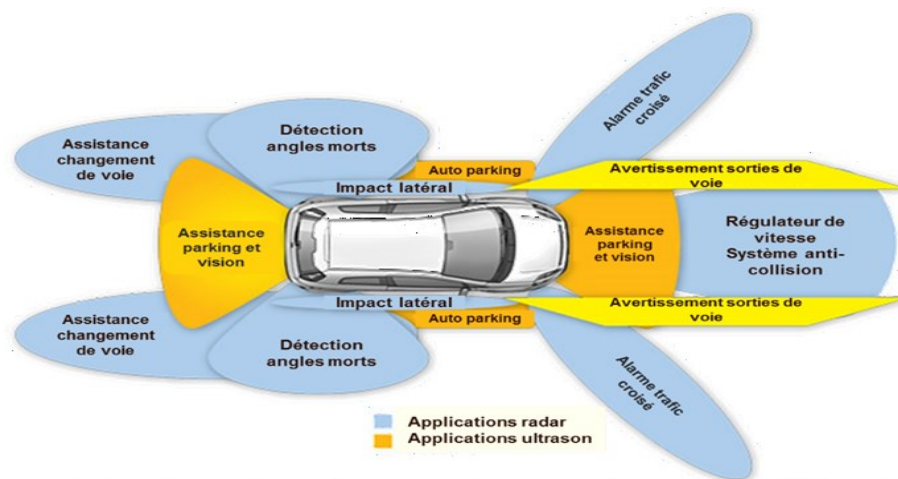


Figure 2-3 Applications des différents capteurs dans un VA [35]

2.1.3 Les véhicules connectés

De nos jours, nous sommes constamment connectés en utilisant des technologies de communication sans fil, que ce soit à la maison, au travail, même en voiture. Des technologies qui conduisent à l'émergence d'une nouvelle génération de véhicules intelligents capables de communiquer entre eux et avec leur environnement. Le partage d'informations à travers les communications véhiculaires sans fils améliore considérablement la sécurité et le confort de tous les passagers dans une portée de plusieurs centaines de mètres ou même plus si on utilise des multi-hop. Il est prévu que les opérations de diffusion d'informations de sécurité, comme la vitesse, l'intention de changement de ligne, etc., constitueront un élément clé dans l'utilisation des réseaux véhiculaires Ad-hoc, en anglais Vehicular Ad-hoc NETworks (VANET).

2.1.3.1 Les réseaux véhiculaires (VANETs)

Les VANETs sont des réseaux mobiles Ad-hoc où des véhicules, des unités routières (RSU : Road Side Unit) et des infrastructures de relais échangent mutuellement des informations, telles que les avertissements de sécurité et les informations sur le trafic qui peuvent être efficaces pour éviter les accidents et les embouteillages. Les trois types de nœuds sont dédiés de dispositifs de communications à courte portée (DSRC : Dedicated Short Range Communications). DSRC travaille à 5,9 GHz avec une bande passante de 75 MHz et une portée approximative de 1000 m. Les communications des véhicules sont habituellement développées dans le cadre de systèmes de transport intelligents (STI) [42]. Ces réseaux ont ouvert les portes d'une part à l'émergence d'applications de sécurité classique telle que : le système de surveillance d'angles morts, le régulateur de vitesse adaptatif, etc. et d'autre part, la création et le développement de nouvelles catégories d'applications telles que : l'infotainment (Information/Entertainment), le paiement à distance, l'optimisation des itinéraires, le contrôle de trafic et pourquoi pas l'automatisation des autoroutes.

Trois modèles de communication ont été développés : Véhicule à véhicule (V2V : Vehicle to Vehicle), véhicule à infrastructure (V2I : Vehicle to Infrastructure) et véhicule à piéton (V2P : Vehicle to Pedertrian), voir Figure 2-4 ci-dessous. Ils regroupent respectivement les systèmes et les mécanismes de communication entre un véhicule et un autre véhicule, un véhicule et une infrastructure et entre un véhicule et un piéton, collectivement appelé véhicule à tout (ou V2X).

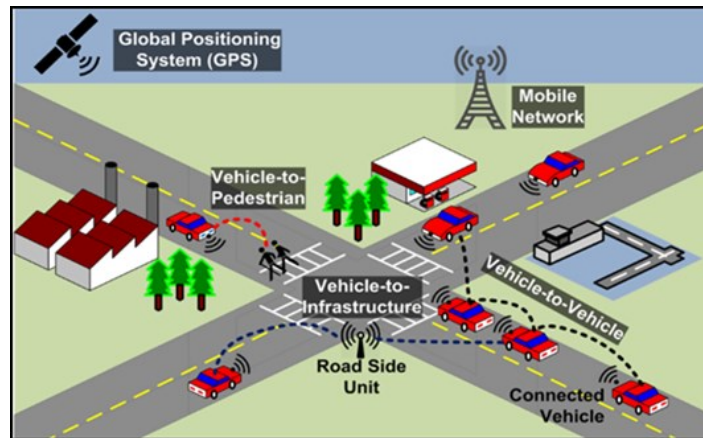


Figure 2-4 Communications V2X (Source : Qatar Mobility Innovations Center (QMIC))

Les équipements utilisés sont : l'unité de côté de la route (RSU : Road Side Unit) qui s'occupe de la collection, la diffusion des données de/aux OBU (en anglais : On-Bord Unit) et de communiquer à travers les autres réseaux (4G/LTE, 5G...) ; l'unité de bord qui s'occupe de la transmission/réception des autres OBU et RSUs ; le GPS qui se trouve à l'intérieur du véhicule.

2.1.3.2 Les standards de communication des VANETs

Présentement deux standards de communication destinée au réseau véhiculaire ont été développés : Wireless Access in Vehicular Environment (WAVE) et Communication Access for Land Mobiles (CALM) qui reste un concept futuriste et loin d'être utilisé dans un futur proche. Ces derniers sont les modes opératoires utilisés par les équipements de communications véhiculaires dans la bande DSRC. WAVE est un système de communication radio destiné à fournir des services de réseaux sans fil interopérable dans des environnements où la couche physique change rapidement (les véhicules), et où le mode d'échange de données de courte durée (de l'ordre de 100 ms) est très nécessaire. Plusieurs groupes de travail (IEEE, ISO, ETSI, etc.) ont spécifié les standards 802.11p et la suite de standards 1609 (1609.1, 1609.2, 1609.3, 1609.4).

2.1.4 Évaluation des risques

Dans le domaine automobile, c'est la méthode ASIL (Automotive Safety Integrity Level), définie dans la norme « ISO 26262 - Véhicules routiers - Sécurité fonctionnelle », qui est souvent utilisée pour l'évaluation et la classification des risques. Comme son nom l'indique, la norme ISO 26262 est une norme internationale pour la sécurité fonctionnelle des véhicules

routiers. Elle est une adaptation de la norme IEC 61508 pour les systèmes électriques/électroniques dans le domaine automobile.

En général, pour définir le niveau de sécurité ASIL et classifier chaque fonction de sécurité, une évaluation des risques est nécessaire. Pour la faire, nous déterminons les niveaux d'exposition, de gravité et de maîtrise des risques pour chaque test (voir figure 2.5).

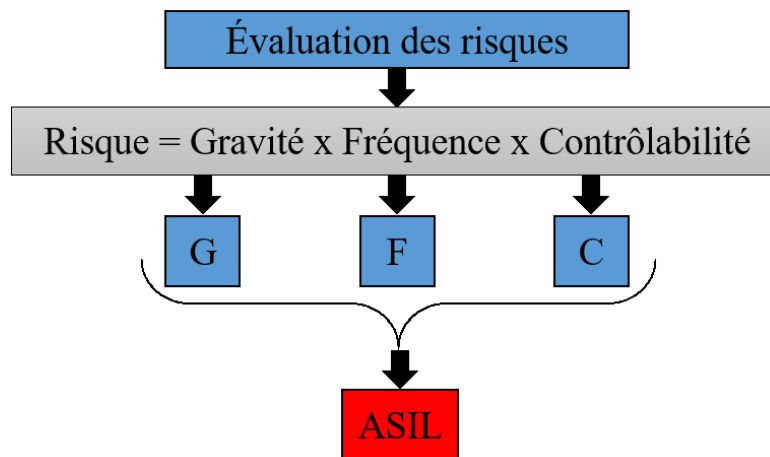


Figure 2-5 Détermination des niveaux ASIL : évaluation des risques

2.2 Approches de test et validation des véhicules intelligents/autonomes

Tel que mentionné précédemment, pour démontrer les performances attendues d'un système avancé d'aide à la conduite dans un véhicule intelligent ou hautement automatisé, les approches de test doivent inclure une combinaison de simulation, de tests sur pistes d'essai et de tests sur route (souvent appelé en anglais : Field Operational Tests (FOT)) [25]. Dans ce qui suit, une étude de l'état de l'art de chaque approche est présentée.

2.3 Évaluation sur pistes d'essai

Chaque pays a ses propres réglementations et législations concernant l'acceptation ou non de l'importation et/ou la mise en vente d'un nouveau produit. Au Canada et aux États-Unis, les fabricants et les importateurs doivent certifier leurs véhicules selon des normes de sécurité véhiculaires: CMVSS au Canada et FMVSS aux États-Unis. En Europe, une entité tierce

effectue une homologation des véhicules fabriqués ou importés. Le programme d'évaluation des véhicules neufs (NCAP) aux États-Unis (US NCAP) et Euro NCAP en Europe publient des procédures de test pour évaluer les performances des véhicules afin d'aider les consommateurs à choisir leur véhicule. La certification et les tests NCAP peuvent être effectués sur des pistes d'essai dans des centres d'essai spécialisés tels que le Centre d'Essai pour Véhicules Automobiles (CEVA) de Transports Canada, géré par PMG Technologies. Ces centres d'essais basent souvent leurs travaux et essais sur des standards et normes issus de différentes organisations de normalisation ou des entités privées officiellement agréées par un état/pays. Ces normes de sécurité sont des règlements écrits en termes d'exigences de performance minimales de sécurité qu'un échantillon de test doit passer. Dans le domaine véhiculaire, nous citons quelques exemples : CMVSS, FMVSS, USNCAP, EURONCAP, ISO et SAE :

- Canada Motor Vehicle Safety Standards (CMVSS): Un ensemble de normes techniques qu'un véhicule doit satisfaire afin d'être certifié à l'importation au Canada. Les normes CMVSS sont maintenues par Transport Canada, un organisme relevant du pouvoir exécutif du gouvernement fédéral du Canada [81].
- FMVSS Federal Motor Vehicle Safety Standards (FMVSS): Un ensemble de normes techniques qu'un véhicule doit satisfaire afin d'être certifié à l'importation aux États-Unis. Les normes FMVSS sont maintenues par National Highway Traffic Safety Administration (NHTSA), un organisme relevant du pouvoir exécutif du gouvernement fédéral des États-Unis [82].
- United States New Car Assessment Program (USNCAP): C'est un programme gouvernemental de sécurité routière chargé d'évaluer les performances de nouveaux modèles d'automobiles contre les diverses menaces de sécurité routière. Les tests New Car Assessment Program (NCAP) et leurs résultats fournissent des informations cruciales pour les consommateurs sur la sécurité relative aux nouveaux véhicules. La visibilité et l'utilisation des informations NCAP par les consommateurs dans leurs décisions d'achat contribuent à la fabrication et l'achat de véhicules plus sécuritaires, attestent de l'importance accrue de NCAP [83].
- European New Car Assessment Program (EURONCAP) : Un organisme international indépendant qui a pour fonction principale d'effectuer des crash tests afin de tester les

capacités dans le domaine de la sécurité passive des véhicules, notamment automobiles. Ces tests représentent, de manière simplifiée, des scénarios d'accidents de la vie réelle qui pourraient blesser ou tuer les occupants d'un véhicule ou d'autres usagers de la route [84], [85].

- International Organization for Standardization (ISO) : Cette organisation créée en 1947 a pour but de produire des normes internationales dans les domaines industriels et commerciaux appelées normes ISO [86].
- Society of Automotive Engineers (SAE) : Pareil que l'ISO, SAE est une organisation de normalisation, mais qui met l'accent principalement sur les industries de transport telles que l'automobile, l'aérospatiale et les véhicules utilitaires [87].

Généralement, les tests sur pistes d'essai sont définis par une série de scénarios de tests (appelé souvent matrice de tests) que chaque véhicule doit passer un par un. Ensuite, les résultats d'essais sont évalués selon des exigences de performances définies préalablement dans le standard ou la norme. Le Tableau 2-2 montre un exemple d'une matrice de tests pour un système AEB chez PMG Technologies.

Tableau 2-2 Exemple d'une matrice de tests pour un système AEB

Scénario	Vitesse véhicule d'essai (km/h)	Vitesse de la cible (km/h)	Conditions de test	# de test
A1. Cible arrêtée	10, 20, 30, 40, 50 (-5 en cas d'impact)	0	Jour	Jusqu'à 17
B1. La cible se déplace à une vitesse constante plus lente	40 (TTC=5s @ 34m)	16	Jour	7
B2. La cible se déplace à une vitesse constante plus lente	72 (TTC=5s @ 56m)	32	Jour	7
C1. La cible décélère jusqu'à un arrêt complet.	56 (14 m en avance)	56 @ 0.3g	Jour	7
G1. Discrimination d'obstacle faible (plaque d'acier)	40, 72	-	Jour	14

2.3.1 Documents pertinents actuels et au niveau mondial relatif à la normalisation des véhicules intelligents/connectés/autonomes

Une liste non exhaustive des différents standards/références actuels des véhicules intelligents/connectés/ autonomes est présentée en Annexe B [25], [41], [88]–[141], [141]–[145], [145]–[155]. En octobre 2018, le ministère des Transports des États-Unis a publié «Préparer l'avenir des transports: véhicules automatisés 3.0» [156]. Le document regroupe aussi dans son annexe les références les plus pertinents au développement des véhicules autonomes regroupés par domaines fonctionnels.

2.4 Évaluation par simulation

Pour l'évaluation des systèmes avancés d'aide à la conduite dans un véhicule intelligent ou hautement automatisé, les tests de conduite réels s'avèrent être la solution la plus fiable. Mais à cause du temps de préparation important et leur cout exorbitant cette solution n'est pas optimale. Ces tests ne peuvent pas couvrir toutes les conditions de fonctionnement et ne peuvent pas être reproduits exactement avec les mêmes conditions et/ou exigences de départ. Un moyen de contourner ce problème consiste à utiliser des outils de validation virtuels.

2.4.1 Pendant la phase de développement

Pendant la phase de développement, les différentes étapes d'un processus de validation sont [6]: Model in the Loop (MIL), Software in the Loop (SIL), Processor in the Loop (PIL) and Hardware in the Loop (HIL). Selon Pawel et al [7], le Model-Driven Development (MDD) et le Model-Based Testing (MBT) sont les approches privilégiées pour le développement de systèmes ADAS. Une approche qui nécessite une compréhension et une connaissance approfondies du fonctionnement du système. Les auteurs dans [8] ont présenté différentes approches X-in-the-loop (XiL) et leur validation par rapport à des tests réels.

Model in the Loop (MIL) : Cette étape est basée sur le modèle du système lui-même. Le test d'un système embarqué au niveau MIL signifie que le modèle et son environnement sont simulés dans un cadre de modélisation sans composants matériels physiques. Par exemple : si un contrôleur est à concevoir à partir de zéro, des modèles du contrôleur et de l'installation sont développés en utilisant Matlab/Simulink, puis une vérification est faite. « In the Loop » est une

terminologie qui fait référence au bouclage du système de commande avec le modèle d'environnement multiphysique.

Software in the Loop (SIL) : Tester et valider un système embarqué sur le niveau SIL signifie que le logiciel embarqué est testé dans un modèle d'environnement simulé, mais sans aucun matériel (à savoir pas de composants mécaniques ou hydrauliques, pas de capteurs, actionneurs). Revenons à notre exemple, une fois que le modèle est vérifié, l'étape suivante est de développer un code logiciel/HDL (Hardware Description Language) en fonction du processeur ou FPGA qu'il est prévu d'utiliser pour la mise en œuvre du matériel final. Ensuite, exécuter les simulations pour vérifier que le code a été bien transcrit de telle sorte que cette dernière n'altère pas les exigences du système. Sinon il faut revenir à l'étape MIL pour apporter les changements nécessaires.

Processor in the Loop (PIL) : Cette étape consiste en l'intégration du code généré dans les systèmes embarqués (ECU) avec un matériel spécifique. Le logiciel embarqué fonctionne sur une carte cible avec le processeur cible ou sur un émulateur du processeur cible. Les tests et la validation au niveau PIL sont importants, car ils peuvent révéler des défauts qui sont causés par le compilateur cible ou par l'architecture du processeur. Donc pour notre exemple, une fois que le code du contrôleur est vérifié en SIL, ce code va être chargé sur le processeur / FPGA pour exécuter les simulations avec le modèle de l'installation afin de valider son bon fonctionnement. Sinon il faut revenir à l'étape SIL ou MIL pour apporter les changements nécessaires.

Hardware in the Loop (HIL) : Une fois que le code généré a été vérifié sur un matériel spécifique en utilisant PIL, maintenant le modèle de l'installation peut être remplacé par le matériel original, mais dans un environnement dit de laboratoire. Pour notre exemple, si le contrôleur a été développé pour un moteur à courant continu, ce contrôleur de vitesse est implémenté en FPGA ou processeur qui est maintenant interfacé au moteur à courant continu en connectant les entrées et les sorties aux bons points de capteurs ou transducteurs et effectuer le test HIL pour validation.

Vehicle Hardware in the Loop (VEHIL) : Pour une méthodologie plus efficace dans le domaine véhiculaire et surtout pour la validation des ADAS et les systèmes de communication intervéhiculaire, un nouveau concept a été proposé (VEHIL) par TNO Automobile [9]–[11]. Cette approche consiste en l'intégration d'un véhicule dans la boucle de validation, mais ça reste

dans un environnement de laboratoire. Le principe de fonctionnement de cette approche de validation est mentionné dans la Figure 2-6. Dans un laboratoire VEHIL, un environnement virtuel est défini dans lequel le véhicule, l'infrastructure et leurs interactions sont simulés en temps réel.

Les avantages d'une telle approche sont énormes : moindre coût, un grand nombre d'essais qui peuvent être effectués dans une courte durée, les tests peuvent être réalisés et reproduits avec une grande précision, diminution des risques liés aux tests, flexibilité et rapidité de mettre en place des scénarios de tests et finalement une grande facilitée d'injection des défauts vu que toutes ces entrées sorties sont liées à l'environnement de simulation.

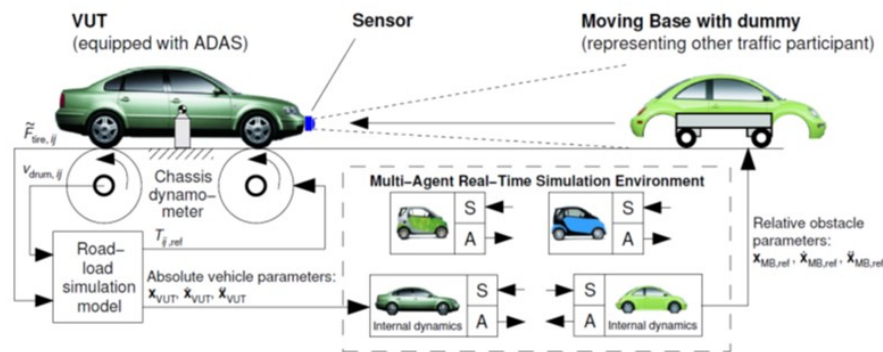


Figure 2-6 Principe de fonctionnement du VEHIL de TNO Automobile [10]

2.4.2 Plateformes de simulation véhiculaire

L'utilisation de plateforme de simulation virtuelle devient un processus inévitable pour le développement et l'évaluation de systèmes dans tous les domaines d'ingénierie. Dans le domaine des transports, leur utilisation dans ce processus rend les systèmes beaucoup plus sécuritaires et fiables, et permet aussi un gain énorme en termes d'argent, temps de préparation et d'intervention. En ce qui suit, nous présentons quelques plateformes de simulation véhiculaire les plus pertinentes. Une comparaison détaillée est présentée dans [40].

PreScan est une plateforme de simulation utilisée dans l'industrie automobile pour le développement de systèmes avancés d'assistance au conducteur (ADAS), qui sont basés sur les technologies de capteurs tels que les radars, lidars, appareils photo et GPS, et les systèmes de sécurité active. PreScan est également utilisé pour la conception et l'évaluation des applications de communication de véhicule à véhicule (V2V) et de véhicule à infrastructure (V2I) ainsi que

des applications de conduite autonomes. PreScan peut être utilisé de la conception de contrôleurs à base de modèle (MIL) jusqu'à des tests en temps réel avec logiciel-in-the-loop (SIL) et hardware-in-the-loop (HIL). Les applications de ce simulateur sont: Autonomous emergency braking, adaptive cruise control, lane departure warning & lane keeping assist, pedestrian detection & avoidance, blind spot warning & lane change assist, intelligent headlights systems, park assist & backup aid, traffic sign recognition, connected driving (V2X Solution: Developing and testing V2X applications) et finalement Automated Driving [65].

Le groupe ESI de son côté gère le logiciel Pro-SiVIC, une autre plate-forme de prototypage virtuel, qui permet d'effectuer des simulations 3D des environnements et des capteurs réalistes, comme ceux utilisés dans les systèmes avancés d'assistance au conducteur (ADAS), y compris les infrastructures, les usagers de la route, la dynamique du véhicule et les différentes technologies de capteur et de perception, dans un environnement entièrement virtuel. Ce logiciel contient des modèles avancés de véhicules, de piétons et d'animation d'objet mobile. Il peut reproduire avec précision des environnements très complexes [66].

Un projet d'interfaçage entre OPAL-RT et PRO-SiVic a été bien réussi. OPAL-RT est une entreprise canadienne connue par le développement de plateformes de simulation en temps réel avec une très grande précision et des capacités de gestion et de traitement de données avec des architectures multiprocesseurs. Elle offre une grande gamme de solutions: RT-LAB entièrement intégré avec MATLAB/Simulink, (RT-LAB: open Real-Time Simulation Software Environment), eHS, SSN, Artemis, RT-LAB Orchestra et Dinamo [67].

Le groupe dSPACE est un autre fournisseur très connu dans le domaine des simulations hardware et software. Il fournit des solutions dans pas mal de secteurs : aérospatiale, machine électrique, ingénierie médicale, automobile, etc. Dans ce qui suit, nous allons citer que les solutions automobiles : V2X Solution (Developing and testing V2X applications), TOOL CHAIN (HIL), Automotive Simulation Models (ASM), MotionDesk [68].

CarSim et TruckSim sont utilisés dans une grande variété de simulateurs de conduite (tel que le simulateur de TOYOTA). Cette plateforme est dédiée pour simuler et analyser le comportement des véhicules à quatre roues en réponse aux entrées de direction, de freinage et d'accélération. Les forfaits de base CarSim ne nécessitent pas d'autres logiciels, bien que la pleine compatibilité

avec d'autres environnements de simulation soit incluse pour MATLAB / Simulink, LabVIEW, ETAS ASCET, et les langages de programmation tels que Visual Basic et C [69].

La gamme de logiciels SCANNER, développé par Oktal, répond aux besoins de simulateurs de conduite grâce à deux variantes spécifiques. Studio SCANeR, pour la recherche et de l'ingénierie : un outil logiciel complet répondant à tous les défis de la simulation de conduite. Loin d'être un outil "boîte noire", il est une véritable plate-forme de simulation évolutive, extensible et ouverte, répondant aux besoins des chercheurs et des ingénieurs. SCANeR DT : pour la formation et la prévention : une solution dédiée aux fabricants de simulateurs de formation pour tout type de véhicule [70].

2.4.3 Approches probabilistes

Les méthodes d'évaluation et de validation de test sur piste d'essai dans le domaine véhiculaire sont pour la plupart déterministes. Cela signifie que plusieurs facteurs de fonctionnement, tels que la vitesse des véhicules, sont traités comme s'ils étaient fixes et précisément connus. Ils tentent de tenir compte de la variabilité et de l'incertitude en utilisant des facteurs de sécurité fixes. Par contre, les approches probabilistes permettent de quantifier les variations et les incertitudes en utilisant principalement des distributions de probabilités plutôt que des valeurs fixes. Dans cette même démarche, quelques initiatives ont vu le jour dans le domaine des véhicules intelligents et autonomes [14]–[17].

2.4.3.1 Simulation Monte-Carlo

Les simulations Monte-Carlo appliquées dans le domaine véhiculaire sont souvent basées sur des modèles stochastiques construits à partir des données provenant de projet FOT [17]. Zhao [17] explique en détaille les méthodes utilisées dans [71] et [72]. La prise de décision et l'intégration numérique stochastique dans le processus d'évaluation des systèmes d'évitement des collisions ont été étudiées par Jansson, Karlsson et al [15], [73]. Le système étudié utilise des capteurs radar pour détecter et suivre d'autres véhicules. Puisque les informations des capteurs inexacts peuvent conduire à des informations d'état incertaines et peuvent influencer la performance du système d'évitement de collision, un algorithme d'évaluation statistique a été développé pour traiter les incertitudes d'estimation en calculant la probabilité pour chaque action.

L'avantage de cette approche est que tous les scénarios et modèles sont extraits des données de projets FOT, donc une bonne interprétation des scénarios de conduite réelle. Les simulations Monte-Carlo peuvent réduire le temps de préparation, le temps d'intervention et le coût de l'évaluation, mais pas nécessairement avec une grande efficacité vu que les parties non critiques de la conduite peuvent dominer les simulations. Pour bien décrire la démarche de simulation Monte-Carlo, Gietelink [74] résume plusieurs travaux et donne les principales étapes de l'utilisation de cette méthode dans l'analyse des systèmes de contrôles.

Supposons que le système en boucle fermée, dans ce cas un véhicule équipé d'un régulateur automatique de vitesse (ACC : Adaptive Cruise Control) et son contrôleur, doit être vérifié pour un certain niveau de performance ρ (exemple : probabilité de dégagement). Le problème consiste alors à déterminer la probabilité p que cette performance ρ soit supérieure à une valeur de seuil préspecifiée γ . Pour calculer $\hat{p}_N(\gamma)$ (la probabilité empirique), nous générons N échantillons indépendants identiquement distribués (iid) q_1, q_2, \dots dans l'espace de paramètres \mathcal{Q} selon sa fonction de densité de probabilité (pdf) f_Q . Le résultat de chaque i -ème expérience est représenté par une fonction indicatrice J , donnée par :

$$J(q_i) = \begin{cases} 0, & \text{if } \rho < \gamma \\ 1, & \text{if } \rho \geq \gamma \end{cases} \quad (2.1)$$

Donc la probabilité empirique $\hat{p}_N(\gamma)$ peut être estimée avec :

$$\hat{p}_N(\gamma) = \frac{1}{N} \sum_{i=1}^N J(q_i) \quad (2.2)$$

Qui est connu comme l'estimateur d'échantillonnage simple et est un estimateur non biaisé, vu que la valeur de son espérance mathématique est égale à celle de p .

$$E\{\hat{p}_N(\gamma)\} = E\left\{\frac{1}{N} \sum_{i=1}^N J(q_i)\right\} = \frac{1}{N} E\left\{\sum_{i=1}^N J(q_i)\right\} = \frac{1}{N} \sum_{i=1}^N E\{J(q_i)\} = \frac{1}{N} Np = p \quad (2.3)$$

La variance σ^2 de \hat{p}_N est donnée par :

$$\begin{aligned} \text{var} \left\{ \frac{1}{N} \sum_{i=1}^N J(q_i) \right\} &= \frac{1}{N^2} \text{var} \left\{ \sum_{i=1}^N J(q_i) \right\} = \frac{1}{N^2} N \text{var} \{ J(q_i) \} = \\ &= \frac{1}{N} \left(E \{ (J(q_i))^2 \} - (E \{ J(q_i) \})^2 \right) = \frac{1}{N} (p - p^2) = \frac{p(1-p)}{N} \end{aligned} \quad (2.4)$$

Puisque le résultat de l'inégalité $|p - \hat{p}_N| \leq \varepsilon$, ε est la précision de l'estimation, est une variable aléatoire, il a une probabilité de réalisation. En introduisant un degré de confiance $1 - \delta$, cette probabilité est définie par :

$$\Pr \{ |p(\gamma) - \hat{p}_N(\gamma)| \leq \varepsilon \} \geq 1 - \delta, \quad \forall \gamma \geq 0, \delta \in (0,1), \varepsilon \in (0,1) \quad (2.5)$$

Comme N est requis pour le calcul de (2.4), il peut être calculé par l'application de l'inégalité de Chernoff. Le nombre d'expériences N doit être choisi de telle sorte que : $2e^{-2N\varepsilon^2} \leq \delta$, donc :

$$N_{ch} \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta} \quad (2.6)$$

Algorithme Monte-Carlo

Étant donné $\varepsilon > 0$, $\delta \in (0,1)$ et $\gamma \geq 0$, cet algorithme donne une estimation $\hat{p}_N(\gamma)$ de $p(\gamma)$ avec une précision $|p - \hat{p}_N| \leq \varepsilon$ et une probabilité d'au moins $1 - \delta$

1. Déterminer le nombre d'échantillons nécessaire : $N = N_{ch} \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$
 2. Tirer N échantillons q_1, q_2, \dots selon $f_Q(Q)$
 3. Calculer la probabilité empirique : $\hat{p}_N(\gamma) = \frac{1}{N} \sum_{i=1}^N J(q_i)$
-

2.4.3.2 Échantillonnage préférentiel (IS : Importance Sampling)

La méthode IS est souvent utilisée pour réduire la variance dans la méthode de Monte-Carlo [75]. L'idée est que l'effet des valeurs prises par une variable aléatoire est plus fort que d'autres sur l'estimateur recherché. Donc la variation de notre estimateur peut être réduite si ses valeurs, qui ont le plus d'effet, se réalisent plus souvent.

Il est donc logique d'accorder plus d'attention aux conditions d'exploitation qui sont plus susceptibles de devenir dangereuses que d'autres (exemple : distance initiale et vitesse relative). Une possibilité est d'utiliser une technique comme l'échantillonnage préférentiel afin d'augmenter le nombre d'occurrences de l'événement dont la probabilité p devrait être estimée [16], [74]. La force réelle de l'IS réside dans sa capacité à estimer précisément les probabilités d'événements rares impliquant une variable aléatoire qui est une fonction de plusieurs autres variables aléatoires [76].

Cette méthode a été détaillée et utilisée dans les travaux de Gietelink [16] et Zhao [77]. Le premier l'a appliqué pour l'estimation des événements dangereux dans le même problème décrit dans la section (2.3.3.2). Le deuxième l'a appliqué pour l'identification des événements dangereux, mais dans un problème de changement de voie. En appliquant un échantillonnage préférentiel, Zhao [77] était capable de proposer une nouvelle approche qui permet l'accélération du processus d'évaluation de performances des véhicules autonomes. Il a démontré que son approche peut accélérer le processus d'évaluation de deux mille à vingt mille fois que les tests opérationnels sur terrain (FOT).

Deux autres approches sont souvent mentionnées dans la littérature, l'échantillonnage préférentiel adaptatif (Adaptive Importance Sampling) et la méthode de rejet (Rejection Sampling). Pour un état de l'art plus en détail, nous proposons les références suivantes [16], [76].

L'algorithme IS est décrit comme suit. Compte tenu l'ensemble de paramètres \mathcal{Q} , supposons que l'on veut estimer une probabilité p définit par :

$$p = \int_{\mathcal{Q}} J(q) f_{\mathcal{Q}}(q) dq = E \{ J(q) \} \quad (2.7)$$

Où nous échantillons q à partir de sa densité de probabilité (PDF) conjointe $f_{\mathcal{Q}}$, notée $q \sim f_{\mathcal{Q}}(q)$.

pour mettre en évidence le sous-ensemble \mathcal{Q}^* qui nous intéresse, il est donc logique de ne pas échantillonner à partir du PDF original, mais plutôt d'utiliser une PDF artificielle qui reflète « l'importance » des événements, puis de pondérer les observations pour obtenir une estimation non biaisée. Nous définissons donc une PDF ξ qui est strictement positive sur \mathcal{Q} . On peut alors réécrire l'équation (2.7) comme suit :

$$p = \int_Q \frac{J(q)f_Q(q)}{\xi_Q(q)} \xi_Q(q) dq = E \left\{ \frac{J(\varphi)f_Q(\varphi)}{\xi_Q(\varphi)} \right\} \quad (2.8)$$

Avec $\varphi \sim \xi_Q$. L'estimateur IS basé sur ξ_Q est :

$$\hat{p}_{N_{IS}} = \frac{1}{N_{IS}} \sum_{j=1}^{N_{IS}} \frac{J(\varphi_j)f_Q(\varphi_j)}{\xi_Q(\varphi_j)} \quad (2.9)$$

Avec $\varphi_1, \dots, \varphi_{N_{IS}}$ sont les N_{IS} échantillons iid. Chaque échantillon φ_j tiré de ξ_Q peut être considéré comme une réalisation $\frac{\xi_Q(\varphi_j)}{f_Q(\varphi_j)}$ des échantillons originaux $q_i \sim f_Q(q)$, qui doivent donc être pondérés avec le facteur $\frac{f_Q(\varphi_j)}{\xi_Q(\varphi_j)}$. La valeur attendue de $\hat{p}_{N_{IS}}$ est p , donc nous avons encore une fois un estimateur non biaisé.

La variance σ_{IS}^2 de l'estimateur IS est donnée par :

$$\text{var} \{ \hat{p}_{N_{IS}} \} = \text{var} \left\{ \frac{1}{N} \sum_{j=1}^N J(\varphi_j) \frac{f_Q(\varphi_j)}{\xi_Q(\varphi_j)} \right\} = \frac{1}{N} \left(E \left\{ J^2(\varphi_j) \frac{f_Q^2(\varphi_j)}{\xi_Q^2(\varphi_j)} \right\} - E \left\{ J(\varphi_j) \frac{f_Q(\varphi_j)}{\xi_Q(\varphi_j)} \right\}^2 \right) \quad (2.10)$$

Algorithme d'échantillonnage préférentiel (IS)

Étant donné $\varepsilon, \delta \in (0,1)$, une mesure de performance ρ , un seuil $\gamma \geq 0$ et la fonction de densité de probabilité réelle (PDF) f_Q pour Q , cet algorithme donne une estimation $\hat{p}_N(\gamma)$ de $p(\gamma)$ avec une précision $|p - \hat{p}_N| \leq \varepsilon$ et une probabilité d'au moins $1 - \delta$.

1. Initialisation : $j = 0$
 2. Déterminer une PDF ξ_Q strictement positive dans Q
 3. $j \leftarrow j + 1$
 4. Tirer un échantillon φ_j selon ξ_Q
-

5. Calculer la probabilité empirique cumulative : $\hat{p}_{N_j} = \frac{1}{N_j} \sum_{j=1}^{N_j} \frac{J(\varphi_j) f_Q(\varphi_j)}{\xi_Q(\varphi_j)}$

6. S'il y a une convergence, la probabilité empirique $\hat{p}_{N_{IS}} = \hat{p}_{N_j}$
Sinon, Aller à l'étape 3

2.4.3.3 Méthode de Monte-Carlo par Chaînes de Markov (MCMC)

La principale propriété des méthodes MC discutées ci-dessus est que dans chaque essai le calcul est basé essentiellement sur des distributions fixes et connues au préalable [78]. Par contre, quand le calcul des probabilités ou des espérances mathématiques est introuvable/difficile par des méthodes analytiques ou d'autres approches numériques, les méthodes MCMC s'avèrent très puissantes. Chaque méthode MCMC se décompose en deux étapes, modélisation et réalisation [79]. Un état de l'art de ces méthodes est bien présenté dans [79] et [80].

Une chaîne de Markov est un processus aléatoire ayant un nombre fini d'états, avec des probabilités de transition sans mémoire et peut être représenté par une matrice de transition :

$$T(x' | x) \equiv \Pr(x \rightarrow x') \quad (2.11)$$

Avec $T(x' | x) \geq 0$, $\sum_{x'} T(x' | x) = 1$ pour chaque x . Soit $p^{(0)}$ une distribution initiale de x , et $p^{(t)}$ est la distribution à l'itération t . $p^{(t)}$ est donné par :

$$p^{(t)}(x') = \sum_x p^{(t-1)}(x) T(x' | x) \quad (2.12)$$

L'analyse en régime transitoire consiste à déterminer le vecteur $p^{(t)}$. Il s'agit donc de décrire l'évolution du processus depuis l'état initial jusqu'à l'itération t . Soit la matrice T qui a les éléments $T(x' | x)$ à la ligne x' et à la colonne x , alors :

$$p^{(t)} = T^t p^{(0)} \quad (2.13)$$

Nous appelons les résultats $X_0 = x_0, X_1 = x_1, X_2 = x_2, \dots$ comme une suite de la chaîne commençant à x_0 . Une distribution stationnaire π d'une chaîne de Markov satisfait

$$\pi(x') = \sum_x \pi(x) T(x' | x) \quad (2.14)$$

Ou, d'une manière équivalente :

$$\pi = T\pi \quad (2.15)$$

On note que cela implique que π est un vecteur propre droit de T avec la valeur propre 1. L'objectif de la méthode MCMC consiste à concevoir T telle que π soit la distribution désirée p .

Dans cette partie, nous avons présenté les bases des méthodes MCMC. Dans la littérature, on peut distinguer deux types de méthodes MCMC [80] : des méthodes qui utilisent des marches aléatoires, nous citons l'algorithme de Metropolis-Hastings et l'échantillonnage de Gibbs comme exemples. Et pour accélérer la convergence d'autres méthodes plus complexes ont été développées, telles que Monte-Carlo hybride et Sur-relaxation successive, introduisent des contraintes sur les parcours. L'avantage principal des méthodes MCMC par rapport aux méthodes MC [78]–[80], c'est qu'elles permettent la génération d'échantillons dans des dimensions plus élevées.

Un résumé d'une étude comparative entre les algorithmes les plus utilisés dans la littérature (échantillonnage préférentiel «Importance Sampling», méthode de rejet «Rejection Sampling», échantillonnage de Gibbs «Gibbs Sampling», échantillonnage de Metropolis-Hastings «Metropolis-Hastings Sampling») est présenté dans la section 3.3 (chapitre 3).

2.5 Évaluation par essais sur route (tests opérationnels sur terrain)

Les tests opérationnels sur terrain (FOT) sont des programmes de tests à grandes échelles visant à une évaluation complète de l'efficacité, la qualité, la robustesse et l'acceptation des utilisateurs des nouvelles technologies utilisées dans le domaine véhiculaire, tels que la navigation, les systèmes de communication, les ADAS et les systèmes coopératifs. Le manuel FESTA définit une opération de tests opérationnels sur terrain (FOT) comme : « *Une étude menée pour évaluer une fonction, ou des fonctions, dans des conditions de fonctionnement normales et dans des*

environnements généralement rencontrés par les véhicules en utilisant des méthodes expérimentales ». Selon [157], « *Le principe de ces projets est d'observer le comportement du conducteur sous l'effet d'un ou plusieurs systèmes. Cela doit se faire le plus discrètement possible, en utilisant les véhicules personnels des participants, et sur une longue période pour éviter tout biais dû à l'expérimentation* ». En Europe plusieurs projets FOT ont été financés et lancés destinés principalement à évaluer l'impact de systèmes matures avant leur mise sur le marché, tels que : EuroFOT (CEESAR et l'IFSTTAR), TELEFOT, FESTA et FOTNET.

2.5.1 Projets FOT des systèmes avancés d'aide à la conduite (ADAS)

Plusieurs grands projets FOT ont été lancés aux États-Unis. L'initiative de Volvo (Volvo Intelligent Vehicle Initiative Field Operational Test) : Pendant cette initiative « Volvo Trucks North America » a testé trois systèmes pour des véhicules utilitaires [158] : Système d'alerte avant collision (Collision Warning System (CWS)), régulateur de vitesse adaptatif (Adaptive Cruise Control (ACC)), système de freinage avancé (Advanced Braking System). Les données ont été recueillies pendant 3 ans à partir d'un système d'acquisition de données embarquées dans chaque véhicule. Les objectifs de ce projet étaient : l'accélération du déploiement de ces systèmes, évaluer et valider les performances de sécurité de ces systèmes dans un environnement réel. Les résultats ont montré que les systèmes CWS, ACC et AdvBS peuvent fournir un très grand avantage en termes de sécurité pour des vitesses supérieures à 86 km/h et pendant les événements de précollision qui se produisent à des vitesses élevées.

L'initiative Mack (Mack Intelligent Vehicle Initiative Field Operational Test) : Cette initiative est axée sur l'essai et l'évaluation d'un système d'alerte de franchissement involontaire de ligne (Lane departure warning (LDW)) pour les gros camions [159]. Les données ont été examinées par un évaluateur indépendant, Battelle Memorial Institute, et les coûts et les avantages du LDW ont été calculés. Les évaluations et la validation du système ont également été menées via des sondages des conducteurs. Le but de ce projet était d'évaluer le système LDW en termes de performance de la sécurité et enfin l'acceptation et l'adaptation des conducteurs. Selon Mack Trucks, l'objectif principal était de voir la viabilité de cette technologie pour leur future gamme de produits et déterminer si le LDW valait l'investissement en utilisant les connaissances acquises à partir de ce projet.

2.5.2 Projets FOT des systèmes de communication véhiculaire (V2X)

Pour les systèmes de communication véhiculaire (V2X), la Commission européenne a été très active dans les activités visant à l'amélioration des aspects de la sécurité routière, l'efficacité de la mobilité routière et de la durabilité des transports par le biais de la promotion de son développement. Des projets comme les systèmes de véhicule à infrastructure coopérative (CVIS) et SAFESPOT ont pris les premières mesures en vue de l'utilisation des communications et des capteurs de pointe pour soutenir les pilotes dans leurs tâches de routine, des étapes qui ont conduit la voie à la génération de services coopératifs que nous connaissons aujourd'hui, en exploitant le concept complet de fournir un cadre d'échange de données complexe entre toutes les entités impliquées dans l'environnement routier [23].

Différents projets FOT, portent sur différents domaines du développement et déploiement des services coopératifs, contribue à valider et à faire avancer les différentes parties de l'architecture des communications intervéhiculaires. EuroFOT était le premier projet FOT à grande échelle, centré sur l'impact de plusieurs systèmes intervéhiculaire sur le comportement des conducteurs et des interactions avec ces systèmes. Une autre initiative pertinente était le projet TeleFOT (Aftermarket FOT et les dispositifs nomades dans les véhicules) projet. L'objectif essentiel de ce dernier est d'enquêter sur l'impact des fonctions apportées pour le soutien du conducteur. Des initiatives plus récentes qui peuvent être mises en évidence comprennent les projets DriveC2X et le Fotsis. DriveC2X se concentre sur des applications basées sur les véhicules comme un prolongement naturel des efforts antérieurs, Fotsis traite spécifiquement l'intégration de données relatives au trafic en temps réel par les opérateurs de la route et d'autres entités de l'environnement de la circulation et favorise la participation active de ces entités dans les services coopératives. Dans [23], d'autres projets ont été bien décrits tels que : COMeSafety, COOPERS, SeVeCom, GeoNet.

Aux États-Unis deux grands programmes, commandité par U.S. Department of Transportation (USDOT) National Highway Traffic Safety Administration et en collaboration avec l'institut de recherche sur les transports de l'Université du Michigan (UM), ont été lancé : Safety Pilot Model Deployment (SPMD) Program et Integrated Vehicle-Based Safety Systems Program (IVBSS). Les deux programmes étaient une initiative de recherche mettant en évidence la mise en pratique des technologies, applications et systèmes de sécurité des véhicules connectés dans les vraies

conditions de conduite [160]. Dans notre étude, nous avons opté pour la base de données Safety Pilot Model Deployment Program (SPMD). Le reste des projets FOT lancés aux États-Unis, non cités ici, ont été présentés plus en détail dans Zhao [17], ci après un tableau récapitulatif :

Tableau 2-1 Principaux projets FOT aux États-Unis [17]

<i>Nom</i>	<i>Responsable</i>	<i>Période</i>	<i>Kilométrage [mile]</i>	<i>Véhicule</i>	<i>Capteur</i>	<i>Conducteurs</i>	<i>Sujet de recherche</i>
<i>100 Car Naturalistic Driving Study</i>	<i>Virginia Tech (VT)</i>	<i>2001- 2009</i>	<i>2,000,000</i>	<i>100 berlines</i>	<i>Caméra</i>	<i>Principaux : 109 Secondaires : 132</i>	<i>Collision arrière</i>
<i>ACAS</i>	<i>UM</i>	<i>2004- 2005</i>	<i>137,000</i>	<i>11 berlines</i>	<i>Caméra Radar</i>	<i>96</i>	<i>Avertissement de collision avant</i>
<i>RDCW</i>	<i>UM</i>	<i>2005- 2006</i>	<i>83,000</i>	<i>11 berlines</i>	<i>Caméra Radar</i>	<i>11</i>	<i>Avertissement de sortie de voie</i>
<i>SeMiFOT</i>	<i>UM</i>	<i>2008- 2009</i>	<i>106,528</i>	<i>10 berlines et 4 camions</i>	<i>Caméra Radar</i>	<i>39</i>	<i>Alerte de collision avant, avertissement de sortie de voie, système d'information sur les angles morts, contrôle électronique de la stabilité, et avertissement de perte de valeur</i>
<i>IVBSS</i>	<i>UM</i>	<i>2010- 2011</i>	<i>Berline : 213,309 Camion : 601,944</i>	<i>16 berlines et 10 camions lourds</i>	<i>Caméra Radar</i>	<i>Berlines : 108 Camions : 18</i>	<i>Avertissement intégré</i>
<i>SPMD</i>	<i>UM</i>	<i>2012- 2014</i>	<i>Plus de 34 million</i>	<i>2800 différents types de véhicules</i>	<i>Caméra DSRC</i>	<i>2 700 chauffeurs bénévoles et plusieurs conducteurs de bus et de camions professionnels</i>	<i>Véhicule connecté</i>
<i>Goolge driverless car</i>	<i>Google</i>	<i>2012- 2016</i>	<i>1.3 million</i>	<i>Au - 50 berlines et SUV</i>	<i>Lidar Caméra Radar</i>	<i>Techniciens de Google et volontaires</i>	<i>Véhicule autonome</i>

2.6 Méthodes d'apprentissage automatique et ensembliste

Tout au long de notre étude, plusieurs techniques d'apprentissage automatique (ML) et ensembliste (EL) ont été appliquées. L'objectif et le choix de chaque technique sera discuté dans la section 6.5.2. Le principe et le développement mathématique de chaque technique peuvent être trouvés plus en détail dans les références suivantes [43]–[64] :

Table 2-3 Méthodes d'apprentissages automatiques et ensemblistes utilisées

Nom de la méthode	ML ou EL	Problème résolu	Références
Régression linéaire (LR)	ML	Régression	[43]
Régression linéaire multiple (MLR)	ML	Régression	[45]
Régression à vecteur de support (SVR)	ML	Régression	[46]–[49]
Régression par processus gaussien (GPR)	ML	Régression	[50]–[53]
Régression logistique (LoR)	ML	Classification	[54]–[56]
Classification naïve bayésienne (NB)	ML	Classification	[58], [59]
Analyse discriminante linéaire (LDA)	ML	Classification	[57]
Analyse discriminante quadratique (QDA)	ML	Classification	[57]
K-plus proche voisin (K-NN)	ML	Classification	[60]–[62]
Machines à vecteurs de support (SVM)	ML	Classification	[57], [63]
Machines à vecteurs de support à noyau (K-SVM)	ML	Classification	[57], [63]
Arbre de décision (DT)	ML	Classification	[57], [63]
Bagging [Bootstrap Aggregating]	EL	Classification	[63]
Bagging [Random Forest]	EL	Classification	[63]
Boosting [AdaBoost (Adaptive Boosting)]	EL	Classification	[63], [64]
Stacking [Standard]	EL	Classification	[63], [64]
Stacking [Cross-Validation]	EL	Classification	[64]
Stacking [Vote majoritaire]	EL	Classification	[64]
K-Means Clustering	ML	Classification	[64]

2.7 Conclusion

Les systèmes avancés d'assistance au conducteur constituent la base du développement des véhicules autonomes et passent par un processus d'évaluation et de validation en plusieurs stades: développement, démonstration et déploiement. Dans ce chapitre, nous avons examiné trois approches d'évaluation des systèmes avancés d'aide à la conduite dans un véhicule intelligent ou hautement automatisé. Les principaux avantages et inconvénients de chaque approche sont résumés comme suit :

Les tests de conduite réels s'avèrent être la solution la plus fiable, mais pas nécessairement la plus optimale. En raison de problèmes de temps et de coûts, ces tests ne peuvent pas couvrir toutes les conditions de fonctionnement. De plus, les scénarios de tests ne peuvent pas être reproduits exactement comme ils se sont déroulés pour la première fois.

Un moyen de contourner ce problème consiste à utiliser des outils d'évaluations virtuels; les simulateurs de véhicules sont un très bon exemple. Par ailleurs, en utilisant des simulations uniquement comme approche de validation est souvent insuffisantes en raison du grand nombre de tests virtuels à effectuer. De plus, ils ne sont nécessairement pas représentative de la vraie réalité qu'un véhicule peut faire face pendant toute sa vie. Par conséquent, une approche probabiliste est nécessaire pour réduire cette contrainte, et pour ce faire, une stratégie d'échantillonnage appropriée est souvent utilisée.

Pour les stades de démonstration et de déploiement, deux approches principales sont utilisées: les essais sur des pistes fermées et les essais sur route. Les tests en circuit fermé sont généralement effectués dans des centres de test spécialisés. Plusieurs procédures et approches ont été proposées pour l'évaluation des ADAS. Malheureusement, elles ne peuvent être appliquées qu'à un niveau d'abstraction élevé dans le processus de développement; ils ne fournissent pas d'exigences objectives ni de critères d'évaluation pour la validation et l'étalonnage d'ADAS et ne prescrivent pas l'utilisation d'outils et de méthodes spécifiques dans l'étape de déploiement.

À notre connaissance, la plupart des études/propositions qui tentent de combiner plusieurs approches sont appliquées pendant le stade de développement. Une nouvelle approche qui combine les trois approches d'évaluation pendant le stade de déploiement est nécessaire.

Chapitre 3

Nouvelle stratégie d'échantillonnage de tests opérationnels sur terrain (FOT) basée sur l'algorithme Metropolis-Hastings

Comme l'a déclaré la NHTSA (National Highway Traffic Safety Administration), afin de démontrer les performances attendues d'un système de véhicule hautement automatisé, les approches de test doivent inclure une combinaison entre simulations, essais sur piste et des tests sur terrain/route. La partie simulation doit être basée sur une approche probabiliste. Pour ce faire, une stratégie d'échantillonnage appropriée est souvent utilisée. Dans ce chapitre, nous proposons une nouvelle stratégie d'échantillonnage basée sur les méthodes de Markov Chain Monte-Carlo (MCMC), utilisant l'algorithme Metropolis-Hastings (MH) pour générer des échantillons à partir de distributions de probabilités de tests opérationnels sur terrain (FOT); « Safety Pilot Model Deployment (SPMD) » dans notre cas. Pour choisir l'écart type optimal (σ) pour la distribution de proposition, utilisé dans l'algorithme MH, une combinaison des résultats du test de Kolmogorov-Smirnov avec le taux d'acceptation de chaque écart type est proposée.

3.1 Introduction

Le but ultime des véhicules intelligents est de rendre la conduite plus facile et plus sécuritaire. La principale question est de savoir comment faire en sorte qu'une telle technologie réponde aux exigences de sécurité, en particulier dans les cas les plus défavorables. Choisir des tests physiques précis pour valider leurs performances est certainement la solution. Cependant, une telle approche est de plus en plus coûteuse et prend énormément de temps de préparation et d'implémentation. C'est pourquoi nous devons étudier de nouvelles solutions (outils de simulation, algorithmes de validation, etc.). Afin de valider le comportement, les performances

et de réduire le nombre de tests physiques de véhicules intelligents/autonomes, il est prouvé que les vrais tests de conduite sont la solution la plus fiable, qui s'avère n'est pas la solution la plus optimale. Pour des raisons de coûts exorbitants reliés à la réalisation de ce genre de tests, ces derniers ne peuvent pas couvrir toutes les conditions de fonctionnement. De plus, les scénarios de test ne peuvent pas être reproduits exactement comme ils se sont déroulés pour la première fois. Un moyen de contourner ce problème consiste à utiliser des outils de validation virtuels; les simulateurs de véhicules en sont un très bon exemple. Par ailleurs, l'utilisation des moyens de simulations tout seuls dans un processus de validation est souvent insuffisante en raison du grand nombre de tests virtuels à effectuer. Par conséquent, une approche probabiliste est nécessaire pour réduire cette contrainte. Pour ce faire, une stratégie d'échantillonnage appropriée est souvent utilisée.

Notre stratégie d'échantillonnage proposée [27] repose sur les méthodes MCMC (Markov Chain Monte Carlo), qui utilisent l'algorithme Metropolis-Hastings pour générer des échantillons à partir de distributions de probabilités de tests FOT. Sur la base de cette stratégie d'échantillonnage, nous avons proposé une nouvelle approche d'évaluation virtuelle, illustrée à la Figure 3-1 (présenté dans le chapitre 6), pour les systèmes de sécurité préventive et active, qui implique de tests opérationnels sur le terrain, une approche probabiliste et des tests en simulation avec un logiciel de simulation véhiculaire (PreScan [65]).

3.2 Description du système ADAS étudié

La description du système nous permettra une compréhension générale de son fonctionnement. Dans un cas idéal, cette description est fournie par le constructeur sous forme d'un document regroupant les différentes spécifications techniques et fonctionnelles du système.

Les fonctions de sécurité préventive se composent de 3 couches : Perception, décision et action; et peuvent être classées selon le type d'action :

1. Fonctions d'avertissement : le conducteur est informé d'une situation dangereuse et est responsable de prendre une mesure appropriée.
2. Fonctions d'assistance / intervention : le système applique des actionneurs de commande du véhicule, mais le conducteur peut intervenir à tout moment.

3. Fonctions de contrôle : le système applique automatiquement des actionneurs de commande de véhicule, et le conducteur n'a pas la possibilité d'intervention.

3.2.1 Principe de fonctionnement d'un système AEB

Le système autonome de freinage d'urgence (AEB) est un système de sécurité proprioceptif qui vise à éviter ou à réduire la gravité des collisions arrières. Le système utilise les informations provenant des capteurs qui surveillent la zone devant la voiture itinérante pour identifier les objets avec un potentiel de collision et appliquer une politique de freinage appropriée.

Les systèmes AEB incluent trois fonctionnalités : la fonction de freinage automatique (AEB), la fonction d'avertissement de collision avant (FCW) et la fonction d'interface homme machine. Les algorithmes d'alertes et de prise de décision implémentés dans les systèmes AEB utilisent souvent des mesures basées sur le temps de collision (TTC : Time-to-Collision), la distance critique de freinage, la décélération requise. Pour étudier les systèmes AEB, de nombreux facteurs doivent être pris en considération tels que : le mode de conduite, le type de la route, l'environnement et les éléments du système AEB.

3.2.2 Fonctionnalités et cas d'utilisations

Dans cette section un aperçu de la fonctionnalité du système est fourni. Ensuite, l'objectif et l'intérêt du système sont décrits, ainsi que le comportement prévu du conducteur. La description des cas d'utilisations peut inclure des descripteurs de situations telles que : le type de la cible, le type de route, les conditions environnementales, la condition du trafic, etc. Le système AEB utilisé dans ce chapitre est le suivant:

- Capteurs : Radars (LRR, SRR).
- Commande : TTC.
- Actionneur : Freins.

Le système AEB étudié utilise deux capteurs radar :

- Radar à longue portée (LRR). Portée = rayon étroit de 150 m (9 degrés).
- Système à courte portée (SRR). Portée = faisceau large de 30 m (80 degrés).

Fonctionnement : Le système commence à avertir le conducteur 2,6s avant l'impact prévu (TTC (Time to Collision) = 2.6s). À TTC = 1,6s, une pression de freinage de 40% est appliquée.

Cela donne un temps supplémentaire pour que le conducteur réagisse et augmente la possibilité d'éviter ou de freiner. Si le conducteur commence à freiner seul, le système l'aidera avec une force supplémentaire, atteignant le maximum de la valeur de freinage possible. Si aucune réaction n'est prise, de $TTC = 0,6s$, la pression de freinage complète est appliquée. Il convient de noter que le freinage autonome n'est activé que lorsqu'un objet en collision est enregistré par les deux capteurs. Le freinage partiel nécessite des informations uniquement à partir du premier capteur, le LRR [65].

3.3 Traitement de la base de données FOT

Les tests opérationnels sur terrain (FOT) sont des programmes de tests à grandes échelles visant à une évaluation complète de l'efficacité, la qualité, la robustesse et l'acceptation des utilisateurs des nouvelles technologies utilisées dans le domaine véhiculaire, tels que : la navigation, les systèmes de communication, les ADAS et les systèmes coopératifs.

Nous avons opté pour le programme "Safety Pilot Model Deployment (SPMD)", une initiative de recherche mettant en évidence, la mise en pratique des technologies, applications et systèmes de sécurité des véhicules connectés "V2V, V2I (DSRC)" dans de vraies conditions de conduite. Ce programme a enregistré la conduite de 2 842 véhicules équipés à Ann Arbor, Michigan pendant plus de deux ans. En avril 2015, 34,9 millions de miles ont été enregistrés, faisant de lui (programme SPMD) une des plus grandes bases de données rendues publiques. Six (6) mois de données "du 01-10-2012 au 30-04-2013" peuvent être récupérés du site de l'USDOT [161].

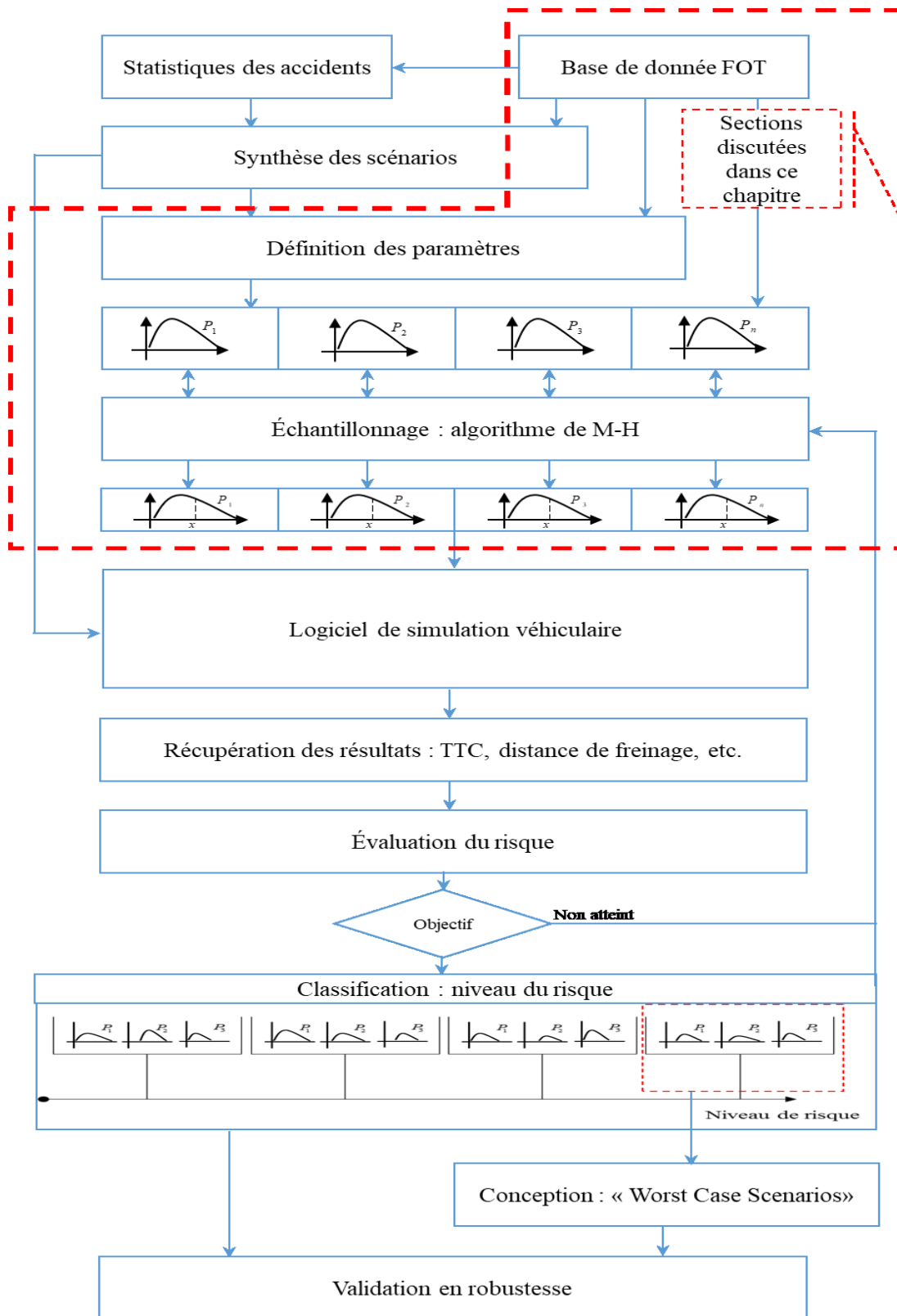


Figure 3-1 Architecture d'évaluation proposée

Comme mentionné dans la Figure 3-2, la base de données SPMD contient 8 ensembles de données [162]:

- Ensemble de données de conduite (Driving dataset)
 - DAS1
 - DAS2
- BSM dataset
- RSE dataset
- Ensemble de données contextuelles (Contextual data dataset)
 - Météo (Weather)
 - Réseau (Network)
 - Programme (Schedule)

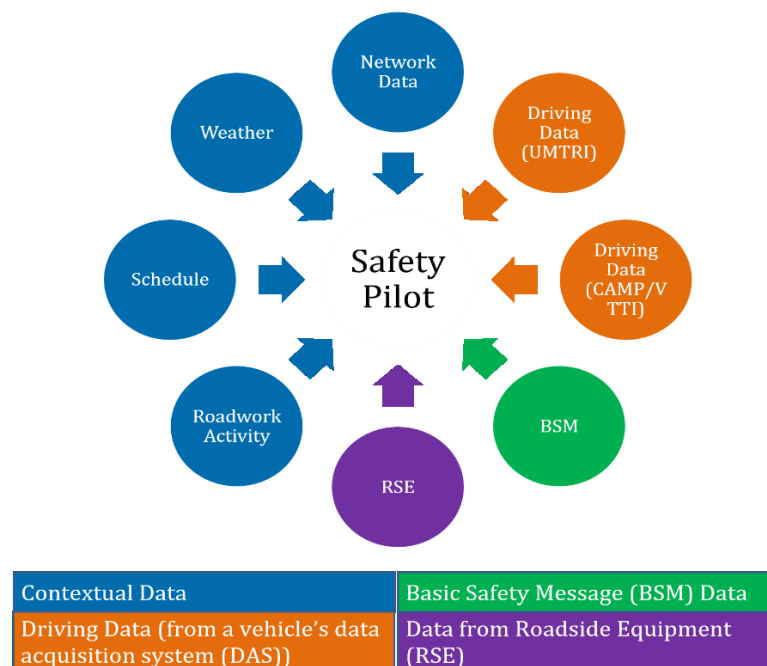


Figure 3-2 Contenu de la base de données SPMD [162]

Trois ensembles de données seront utilisés dans ce chapitre : DAS1, DAS2, Weather.

3.3.1 Données DAS1

Le dataset DAS1 contient des données collectées par le capteur Mobileye, qui fait partie du système d'acquisition de données (Data Acquisition System : DAS), dans chaque véhicule. Une

grande partie de ces données comprend des données sur l'objet devant le véhicule hôte. Les données utilisées dans cette étude sont : Range (distance relative), Rangerate (vitesse relative), Target Type (type de la cible), Status (statut de la cible).

3.3.2 Données DAS2

Le dataset DAS2 contient les données de position, de mouvement et de l'unité de radar des véhicules. Les données utilisées dans cette étude sont : ObjectType (type de la cible), RangeX (distance relative), SpeedX (vitesse relative), TargetMoving (statut de la cible).

3.3.3 Données météorologiques

Les conditions météorologiques ont un impact significatif sur le fonctionnement et les performances des systèmes de transport en général, dans notre cas d'étude, les systèmes d'aide à la conduite. La base de données SPMD inclut les conditions météorologiques, obtenues auprès du National Climatic data Center aux États-Unis, sous lesquelles les véhicules fonctionnaient. Cette base de données comprend : la température, les précipitations, la vitesse du vent, la couverture nuageuse, et lorsque disponible, la température de la surface de la route.

Vu que la base de données SPMD ne contient que deux mois de données météorologiques, nous avons opté pour le site Weather Underground (www.wunderground.com). Un site qui donne accès aux conditions météorologiques de partout dans le monde, ce qui permet l'extraction des conditions météorologiques pour des zones plus spécifiques, « Montréal » dans notre cas.

Les données météorologiques internationales du site Weather Underground [163] sont collectées directement à partir de plus de 29 000 stations météorologiques situées dans des pays dans le monde entier, y compris :

- Environ 6 000 stations météorologiques automatisées fonctionnant dans les aéroports. Généralement, ces stations appartiennent à des organismes gouvernementaux et aéroports internationaux et les données sont mises à jour à intervalles de 1, 3 ou 6 heures, selon la station.
- Plus de 8 000 stations météorologiques personnelles (PWS) et 16 000 stations MADIS (Meteorological Assimilation Data Ingest System [164]).

Les données utilisées dans cette étude proviennent de la station météorologique de l'aéroport international Montréal-Mirabel (CYMX) entre la période du 1er janvier 2007 au 31 décembre 2016.

Par rapport à d'autres technologies de détection, les radars sont connus par leurs robustesses dans des environnements difficiles (mauvaise lumière, etc.) et leurs indépendances aux conditions météorologiques [165]. Dans notre cas d'étude, nous supposons que les conditions météorologiques auront une influence plus significative sur le comportement du véhicule (glissement, etc.) que sur les radars LRR et SRR utilisés par le système AEB étudié.

3.4 Estimation des modèles de fonctions de densités de probabilités (PDF)

Pour prouver la validité de la stratégie d'échantillonnage proposée, nous l'appliquons sur un scénario de conduite avec une configuration de deux véhicules, un véhicule cible à l'arrêt et un autre mené d'un système AEB et qui se rapproche à différentes vitesses et distances (voir Figure 3-3). Les données à extraire sont : Distance relative, vitesse relative et les conditions météorologiques (Température, Humidité, Visibilité, Évènement [Ensoleillé, Pluie, Brouillard, Neige]).

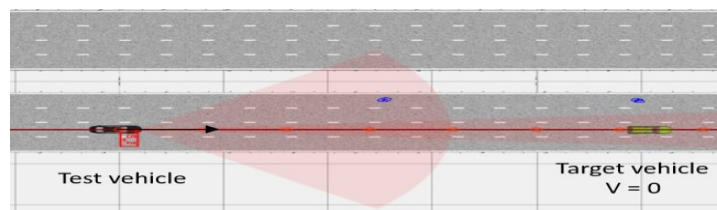


Figure 3-3 Scénario de test étudié

Afin d'extraire que les données liées aux mêmes conditions de ce scénario, deux conditions de recherche ont été appliquées :

- Type de la cible : véhicule.
- Vitesse de la cible égale à zéro.

Pour estimer les modèles des PDF de chaque paramètre, nous avons appliqué deux méthodes d'estimations : une estimation par noyau (Kernel density estimation) et une autre estimation

basée sur l'algorithme EM (Expectation-Maximization algorithm). Les explications et les résultats des deux méthodes seront discutés plus en détail dans le chapitre 5, sections 5.6.4 et 5.12.1 respectivement.

3.5 Implémentation de l'algorithme : Metropolis-Hastings

Actuellement, les méthodes d'échantillonnage largement utilisées dans la littérature sont les méthodes de Monte-Carlo (MC). Le principal inconvénient des méthodes MC par rapport aux méthodes de Markov Chain Monte-Carlo (MCMC) est qu'elles ne permettent pas la génération d'échantillons dans des dimensions plus élevées [80]. On distingue deux types de méthodes MCMC: des méthodes qui utilisent des marches aléatoires. Nous citons l'algorithme Metropolis-Hastings et l'échantillonnage de Gibbs à titre d'exemple. Et des méthodes qui ont été développées pour accélérer la convergence, telle qu'« Hybrid Monte Carlo » et « Successive Over-relaxation ».

Les méthodes MCMC fournissent une séquence d'échantillons à partir d'une distribution de probabilité pour laquelle l'échantillonnage direct est difficile. Le but est de concevoir une chaîne de Markov de sorte que la distribution stationnaire de cette chaîne soit exactement la distribution que nous voulons échantillonner [166]. Notre stratégie d'échantillonnage proposée est basée sur l'algorithme Metropolis-Hastings. Le choix de l'algorithme a été fait après une étude comparative entre les algorithmes les plus utilisés dans la littérature (échantillonnage préférentiel « Importance Sampling », méthode de rejet « Rejection Sampling », échantillonnage de Gibbs (en anglais : Gibbs Sampling), échantillonnage de Metropolis-Hastings « Metropolis-Hastings Sampling »). Un résumé de l'étude est présenté dans Table 3-1 et Table 3-2.

Table 3-1 Méthodes MC en un coup d'œil

MC	
Échantillonnage préférentiel	Méthode de rejet
Estimation	Échantillonnage
Choix des distributions basé sur l'intuition	
Moins performant dans des dimensions élevées	

Facile à comprendre et à implémenter

Table 3-2 Méthodes MCMC en un coup d'œil

MCMC	
Échantillonnage de Gibbs	Échantillonnage de Metropolis-Hastings
Échantillonnage	Échantillonnage
Nécessite peu de connaissances sur la géographie des distributions	
Plus performant dans des dimensions plus élevées	
Très lent et nécessite des connaissances sur les probabilités conditionnelles	Facile à implémenter, mais nécessite une grande puissance de calcul

Les explications et les résultats de l'implémentation de l'algorithme seront discutés plus en détail dans le chapitre 6, sections 6.7 et 6.12.2 respectivement.

3.6 Choix de l'écart type optimal (σ)

Pour choisir l'écart type optimal (σ) pour une distribution de proposition, nous appliquons un test de Kolmogorov-Smirnov. Le test de Kolmogorov-Smirnov est un test d'hypothèse non paramétrique souvent utilisé pour évaluer la différence entre les fonctions de distribution cumulative (CDF) des distributions de probabilité données, et il est défini par:

$$d_U(F, G) = \sup_x |F(x) - G(x)|, \quad x \in \mathbb{R} \quad (3.1)$$

Où $F(x)$ et $G(x)$ sont les CDF des distributions cible et de l'échantillon respectivement dans notre cas.

Nous proposons de combiner les résultats du test de Kolmogorov-Smirnov avec le taux d'acceptation de chaque écart-type pour choisir la valeur optimale. Ici est un résumé de la stratégie proposée:

- 1) Poser $\sigma = 1$
- 2) Répéter
 - Appliquer l'échantillonneur de Metropolis-Hastings
 - Appliquer le test de Kolmogorov-Smirnov

- Calculer le taux d'acceptation : $Tr = \frac{\text{Nombre d'échantillons acceptés}}{\text{Nombre d'itérations}} \times 100\%$
 - $\sigma = \sigma + 1$
- 3) Jusqu'à $\sigma = M$ (Ici nous avons choisi $M=200$, voir figure)
- 4) Choisir l'écart type (σ) optimal : la première valeur qui passe le test de Kolmogorov-Smirnov et produit le meilleur taux d'acceptation.

Comme exemple, nous prenons la densité de probabilité de la vitesse relative (voir sections 5.4 et 6.7). Pour vérifier l'efficacité de l'échantillonneur de Metropolis-Hastings et la stratégie d'échantillonnage proposée, nous allons discuter trois valeurs différentes de (σ), où nous exécutons 2000 itérations pour chaque valeur. La Figure 3-4 montre que le taux d'acceptation est bien meilleur avec des écarts types (σ) plus faibles, mais une discussion plus détaillée doit être menée pour choisir celui optimal.

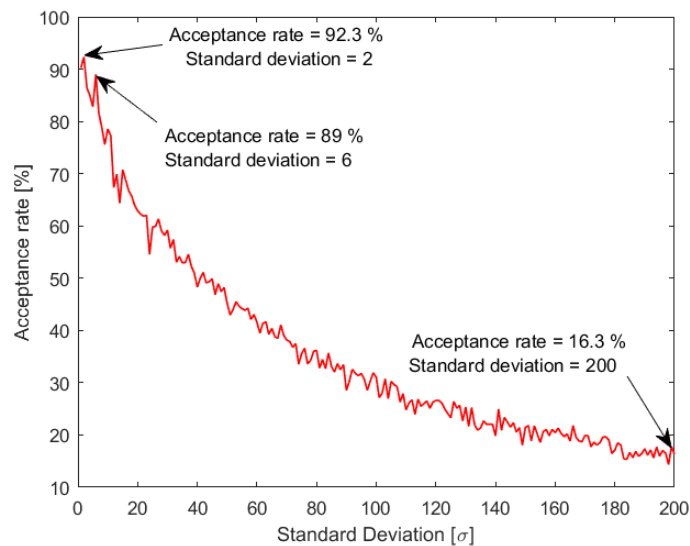


Figure 3-4 Taux d'acceptation en fonction de l'écart type (σ) [Vitesse relative].

Dans un premier cas, nous allons définir une valeur trop basse de l'écart type ($\sigma = 2$) et dans un autre, nous allons le définir trop élevé ($\sigma = 200$), mais dans un troisième cas, nous allons le définir à peu près juste ($\sigma = 6$). Les résultats d'échantillonnage pour le paramètre de vitesse relative sont représentés sur les Figure 3-5, Figure 3-6 et Figure 3-7. Pour les trois valeurs de σ , nous avons deux graphiques. Le premier affiche la vraie distribution cible (en rouge), ainsi qu'un histogramme et sa distribution PDF (en vert) montrant la distribution des échantillons obtenus à l'aide de l'échantillonneur Metropolis-Hastings. Le panneau inférieur représente la chaîne de

Markov réelle: la séquence des valeurs générées. Sur la Figure 3-5, nous voyons ce qui se passe lorsque la distribution de la proposition est trop étroite ($\sigma = 2$), le taux d'acceptation est très élevé (92,3%), de sorte que, techniquement, la chaîne ne reste pas bloquée à un endroit donné. Cependant, on voit l'échantillonneur reste bloquée dans une seule région (entre 100 et 120 km/h) et ne couvre pas toute la plage de fonctionnement.

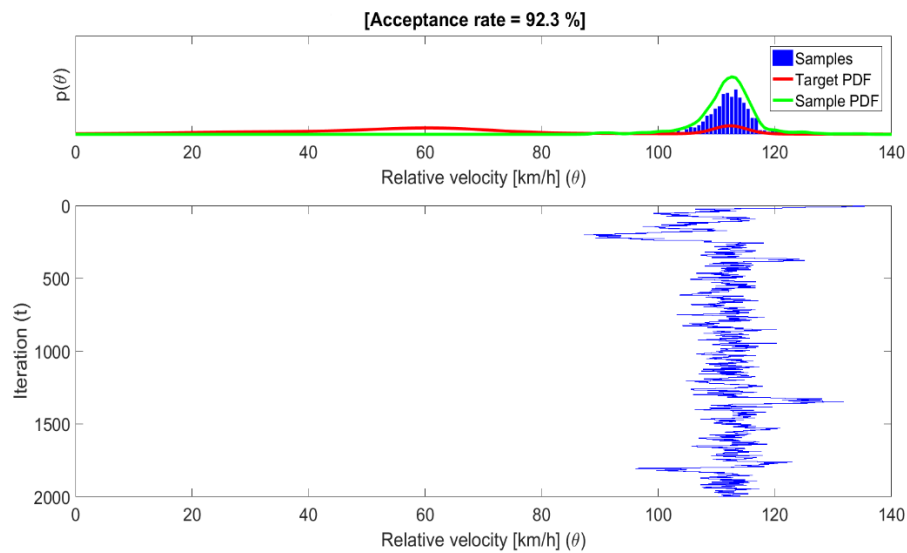


Figure 3-5 Résultats d'échantillonnage ($\sigma = 2$) : Vitesse Relative [2000 itérations]

Sur la Figure 3-6, nous voyons ce qui se passe lorsque nous choisissons une bonne distribution de proposition ($\sigma = 6$): la chaîne montrée dans le panneau inférieur se déplace rapidement sur toute la distribution et couvre toute la plage de fonctionnement, sans rester bloqué à aucun endroit, le taux d'acceptation est de 89%.

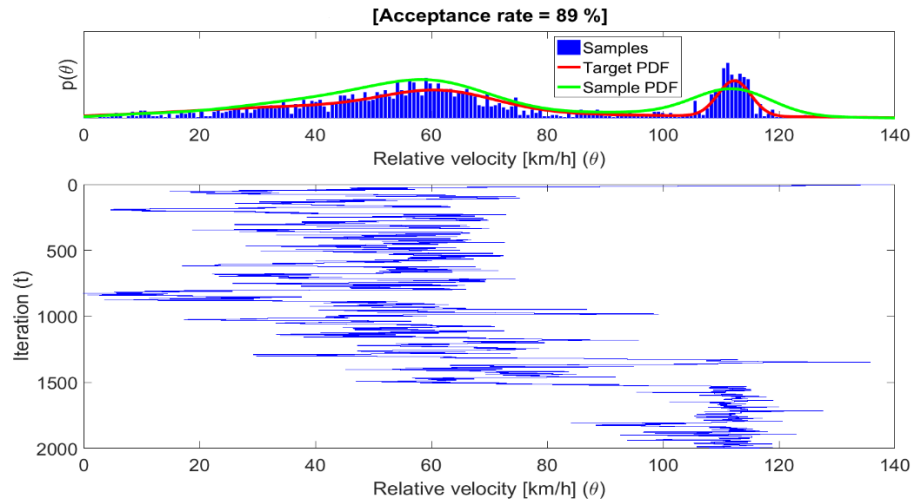


Figure 3-6 Résultats d'échantillonnage ($\sigma = 6$) : Vitesse Relative [2000 itérations]

Enfin, sur la Figure 3-7, si nous définissons la distribution de proposition trop large ($\sigma = 200$): la chaîne parvient à faire de grands sauts, couvrant toute la gamme, mais parce que le taux d'acceptation est si bas (16%), la distribution est très irrégulière.

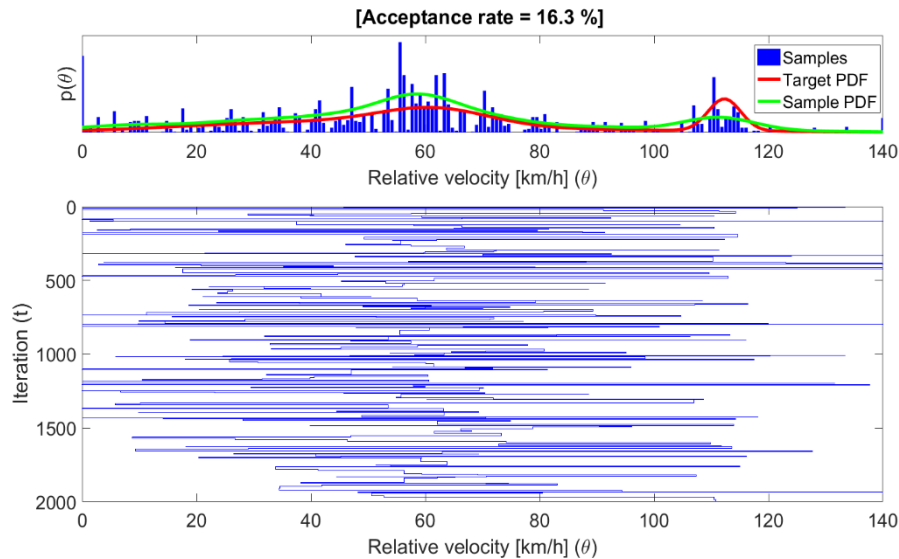


Figure 3-7 Résultats d'échantillonnage ($\sigma = 200$) : Vitesse Relative [2000 itérations]

Pour une décision finale, nous combinons les résultats du test de Kolmogorov-Smirnov avec le taux d'acceptation de chaque écart type afin de choisir une valeur optimale ($\sigma = 6$ pour la vitesse relative, comme indiquée dans le Table 3-3).

Table 3-3 Tableau récapitulatif: écart type optimal (σ)

σ		1	2	3	4	5	6	7	20	80	100	150	200
Taux d'acceptation (%)		90.2	92.3	86.4	84.9	82.8	89.0	81.5	62.9	36.2	31	21.4	16.3
Test de Kolmogorov-Smirnov	h	1	1	1	0	0	0	0	0	0	0	0	0
	p	5^E-5	6^E-3	0.03	0.05	0.48	0.72	0.82	0.97	0.92	0.99	0.97	1
	k	0.74	0.66	0.13	0.12	0.06	0.04	0.02	0.01	0.02	0.005	0.01	0

Où h est la décision de test de Kolmogorov-Smirnov ($h = 1$, cela indique le rejet de l'Hypothèse nulle). p est la valeur asymétrique (p -value) du test et k est la statistique du test. Nous voyons que la première valeur qui passe le test Kolmogorov-Smirnov test est $\sigma = 4$, mais puisque le taux d'acceptation de l'écart type $\sigma = 6$ est plus élevé, cette valeur a été choisie.

3.7 Résumé des résultats de la stratégie d'échantillonnage proposée

Nous avons appliqué la même approche avec les densités de probabilité estimées dans le chapitre 5, sections 5.6.4 et 5.12.1 (Distance Relative, Température, Humidité et Visibilité). Nous avons obtenu l'écart type optimal pour chaque paramètre comme suit: $\sigma = 4$ pour la Distance Relative, $\sigma = 5$ pour la Température, $\sigma = 6$ pour l'Humidité et $\sigma = 4$ pour la Visibilité. Les résultats sont illustrés dans l'annexe du chapitre 5 [Figure 5-18 Sampling results ($\sigma = 4$): Relative Distance (2000 iterations); Figure 5-19 Sampling results ($\sigma = 5$): Temperature (2000 iterations), Figure 5-20 Sampling results ($\sigma = 6$): Humidity (2000 iterations); Figure 5-21 Sampling results ($\sigma = 4$): Visibility (2000 iterations)].

Nous pouvons voir que les chaînes représentées dans les panneaux inférieurs se déplacent rapidement sur toute la distribution et couvrent toute la plage de fonctionnement, sans se bloquer dans un seul endroit, les taux d'acceptation sont: 92,2%, 87,35%, 86,55%, 87,45% pour la Distance Relative, la Température, l'Humidité et la Visibilité respectivement. Les résultats donnent une bonne idée sur comment un bon choix de l'écart type peut faire une énorme différence.

3.8 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle stratégie d'échantillonnage d'une base de données de tests opérationnels sur terrain basée sur l'algorithme Metropolis-Hastings et le test de Kolmogorov-Smirnov. La qualité des résultats de la stratégie d'échantillonnage dépend principalement de la qualité de l'estimation des PDF et de l'optimisation de l'initialisation de l'algorithme Metropolis-Hastings. Les résultats de simulation préliminaire ont montré l'efficacité de la stratégie d'échantillonnage proposée avec des taux d'acceptation élevés et une très bonne couverture de l'ensemble des distributions. L'inclusion de l'algorithme Metropolis-Hastings nous a permis de créer une nouvelle stratégie d'échantillonnage facile à mettre en œuvre, robuste et qui garantit quasiment une convergence vers la distribution exacte et, enfin, une approche globale avec une couverture pondérée sur tout l'espace opérationnel.

L'échantillonneur de Metropolis-Hastings n'est applicable que sur des distributions continues, et pour l'appliquer sur des distributions discrètes, telle que la distribution du paramètre événement [Ensoleillé, Pluie, Brouillard, Neige] (voir Figure 4-3, chapitre 5, section 5.5), nous avons proposé une modification de l'algorithme Metropolis-Hastings que nous appelons « Modified Metropolis-Hastings Algorithm for Efficient Sampling from Discrete Probability Distributions (MMH-DPD) [26] ». Une description plus en détail est présentée dans le prochain chapitre.

Chapitre 4

Modified Metropolis-Hastings Algorithm for Efficient Sampling from Discrete Probability Distributions (MMH-DPD)

4.1 Avant-propos

4.1.1 Auteurs et affiliation

Nacer Eddine Chelbi : Candidat Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Membre du Laboratoire sur l'Intelligence Véhiculaire (LIV).

Denis Gingras : Professeur, Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Responsable du Laboratoire sur l'Intelligence Véhiculaire (LIV).

Claude Sauvageau : Directeur essais et immobilisations, PMG Technologies inc. Blainville, Canada.

4.1.2 Date de soumission

08 juin 2018.

4.1.3 État de l'acceptation

Version finale publiée.

4.1.4 Conférence

IEEE International Conference on Intelligent Transportation Engineering (ICITE)

4.1.5 Référence

N. E. Chelbi, D. Gingras, and C. Sauvageau, “Modified Metropolis-Hastings Algorithm for Efficient Sampling from Discrete Probability Distributions (MMH-DPD) Applied to Field Operational Tests database (SPMD),” in 2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE), 2018, pp. 125–131.

4.1.6 Titre français

Algorithme de Metropolis-Hastings modifié pour un échantillonnage efficace à partir de distributions de probabilités discrètes (MMH-DPD)

4.1.7 Résumé français

L'algorithme Metropolis-Hastings (M-H) est la méthode de Markov Chain Monte-Carlo (MCMC) la plus répandue. Essentiellement, l'algorithme M-H génère un échantillon, l'accepte ou le rejette en fonction d'une probabilité d'acceptation liée à une distribution de probabilité cible continue. Dans ce chapitre, nous proposons un algorithme modifié de Metropolis-Hastings (MMH-DPD) pouvant tirer des échantillons à partir de distributions de probabilité discrètes. Pour commencer, la distribution de probabilité discrète est remplacée par une distribution multimodale et une nouvelle étape après l'étape de rejet et d'acceptation est ajoutée à l'algorithme original. Pour réduire l'erreur causée par la queue de la distribution multimodale, nous avons utilisé un mélange de Gaussiennes généralisées. Des résultats numériques et une généralisation de l'algorithme proposé sont fournis. Nos simulations montrent que l'échantillonneur proposé crée de manière fiable une chaîne de Markov générant une séquence de valeurs, de sorte que, lorsque le nombre d'échantillons augmente à l'infini, nous pouvons garantir qu'ils reflètent les échantillons de la distribution discrète cible.

4.2 Abstract

Metropolis-Hastings algorithm (M-H) is the most popular Markov Chain Monte Carlo (MCMC) method. Essentially, the M-H algorithm generates a sample, accepts or rejects the sample based on an acceptance probability that is related to the continuous target probability distribution. In this chapter, we propose a modified Metropolis-Hastings algorithm (MMH-DPD) that can draw samples from discrete probability distributions. For starters, the discrete probability distribution is replaced with a multimodal distribution and a new step after the rejection and acceptance step is added to the original algorithm. To reduce the error caused by the tail of the multimodal distribution, we used a mixture of Generalized Gaussians instead. Numerical results and a generalization of the proposed algorithm are provided. Our simulations show that the proposed sampler reliably creates a Markov chain that generates a sequence of values, in such a way that as the number of samples goes to infinity, we can guarantee that they reflect samples from the target discrete distribution.

4.3 Introduction

Two ways of thinking about statistics are used for performing inference in many scientific fields: a frequentist approach where parameters are treated as fixed but unknown quantities and the Bayesian approach where parameters are treated as random variables that can be described with probability distributions. Over the last years, Bayesian methods have become very popular in machine learning, digital communications, signal processing, etc. For posterior inference, various deterministic algorithms are used, such as Variational Bayes, Laplace Approximation, etc. However, these methods can be rather complicated to derive and limited in their domain of applicability [167]. To resolve this problem, an alternative class of algorithms based on the idea of Monte Carlo approximation is used.

Monte Carlo simulations try to take advantage of modern computers. The idea is to input values into complex dynamic systems and watch the result over tens of thousands of iterations. In the vehicular domain, this technique is often based on stochastic models constructed from Field Operational Tests (FOT) databases. Zhao [17] explains in detail the application of this technique in [71] and [72]. Decision-making and stochastic numerical integration in the evaluation process

of collision avoidance systems have been studied by Jansson in [15] and [73]. To describe the Monte Carlo simulations technique, Gietelink [74] summarizes several works and gives the main steps of this method applied in control systems analysis. Monte Carlo simulations can reduce the preparation time and cost of an evaluation process, but not necessarily with great efficiency, since the non-critical events of driving can dominate the simulations.

To deal with this inconvenient, the Importance Sampling (IS) technique is often used. We know that the effect, on the desired estimator, of some events taken by a random variable is stronger than others. Therefore, the variation of our estimator can be reduced if these events, which have the most effect, are realized more often. It is, therefore, more convenient to pay more attention to operating conditions that are more likely to become dangerous than others. One possibility is to use the Importance Sampling technique to increase the number of occurrences of these events whose probabilities should be estimated [16], [74]. The real strength of the IS technique lies in its ability to accurately estimate the probabilities of rare events involving a random variable that is a function of several other random variables [76]. This method has been discussed and used in [16] and [77] work, where dangerous events probabilities were estimated in an adaptive cruise control (ACC) and lane change problems successively. The inconvenience of the MC methods is that they do not work well in high-dimensional spaces [80]. The most popular method for sampling from high-dimensional distributions is Markov chain Monte Carlo or MCMC [168]. Two types of MCMC methods can be distinguished: methods that use random walks, we quote Metropolis-Hastings algorithm and Gibbs sampling as examples. Moreover, methods that have been developed to accelerate convergence, such as Hybrid Monte Carlo and Successive Over-relaxation. A state of the art of these methods is well presented in [80]. The most popular MCMC method is undoubtedly the Metropolis–Hastings (M-H) algorithm [169]. Essentially, the M-H algorithm generates a sample, accepts or rejects the sample based on an acceptance probability that is related to the continuous target probability distribution.

Generating random values from finite and discrete distribution is an important block in computer simulations. In this chapter, we propose a modified Metropolis-Hastings algorithm (MMH-DPD) based on a mixture of Generalized Gaussians that can draw samples from discrete probability distributions. This provides an efficient sampling technique in such a way that as the

number of samples goes to infinity, we can guarantee that they reflect samples of the discrete target distribution.

The structure of this chapter is as follows. Section 4.4 describes the original Metropolis-Hastings algorithm. The proposed algorithm, MMH-DPD, is presented in section 4.5 using a numerical example. Section 4.6 presents a generalization of the proposed algorithm.

4.4 Original Metropolis-Hastings Algorithm

The original Metropolis-Hastings algorithm aims to sample from a continuous probability density $p(\theta)$. The Metropolis sampler creates a Markov chain that generate a sequence of θ values, denoted $(\theta^{(1)}, \theta^{(2)}, \theta^{(3)}, \dots)$, in such a way that as $n \rightarrow \infty$, we can guarantee that $\theta^{(n)} \sim p(\theta)$:

$$\theta^{(1)} \rightarrow \theta^{(2)} \rightarrow \dots \rightarrow \dots \quad (4.1)$$

Where $\theta^{(t)}$ is the Markov chain state at iteration t . After the burn-in phase, the samples in the chain begin to reflect samples of the target distribution $p(\theta)$.

We initialize the algorithm with a random value $\theta^{(1)}$. We then use a proposition distribution $q(\theta|\theta(t-1))$ to generate a candidate θ^* . There are a few minor technical constraints to be taken into consideration before choosing the proposition distribution, but for the most part, it can be anything that we like; which makes this algorithm a completely flexible method. A normal distribution centred on the current state is often used $q(\theta|\theta(t-1)) = N(\theta^{(t-1)}|\sigma^2)$; this is called the Random Walk Metropolis-Hastings algorithm (or the Metropolis Algorithm), and this is what we are going to use in this article.

The next step is to accept or reject the proposal. The probability of accepting the proposal is as follows:

$$\alpha = \min\left(1, \frac{p(\theta^*)}{p(\theta^{(t-1)})}\right) \quad (4.2)$$

To decide on accepting or rejecting the proposal, we generate a uniformly distributed number μ between 0 and 1. If $\mu \leq \alpha$, we accept the proposition and the next state is equal to the

proposition: $\theta^t = \theta^*$. If $\mu > \alpha$, we reject the proposition, and the next state is equal to the current state: $\theta^t = \theta^{(t-1)}$. We continue to generate new proposals and accept or reject until the sampler reaches convergence. Here is a summary of the Metropolis sampler steps [169]:

Original Metropolis-Hastings algorithm

7. Set $t = 1$
 8. Generate an initial value μ and θ , and set $\theta^t = \mu$
 9. Repeat
 - a. $t = t + 1$
 - b. Generate a proposition θ^* from $N(\theta^{(t-1)} | \sigma^2)$
 - c. Compute acceptance probability: $\alpha = \min\left(1, \frac{p(\theta^*)}{p(\theta^{(t-1)})}\right)$
 - d. Generate μ from a uniform distribution: $Uniform(0,1)$
 - e. If $\mu \leq \alpha$, accept the proposition and set $\theta^t = \theta^*$, else set $\theta^t = \theta^{(t-1)}$
 10. Until $t = T$
-

4.4.1 Example: Relative Velocity distribution

We apply the Metropolis-Hastings algorithm to a relative velocity distribution $p(x)$ as shown in Figure 4-1, presented by equation (4.3), with $0 < x < 140$ km/h.

The distribution has been retrieved from the ‘‘Safety Pilot Model Deployment (SPMD)’’ database. SPMD is a research initiative highlighting the application of connected vehicle safety technologies and 'V2V, V2I (DSRC)' systems in real driving conditions. This program recorded 2,842 equipped vehicles in Ann Arbor, Michigan for more than two years. In April 2015, 34.9 million miles were recorded, making it one of the largest databases made public [17].

$$P(x) = a_1 e^{-\left(\frac{(x-b_1)}{c_1}\right)^2} + a_2 e^{-\left(\frac{(x-b_2)}{c_2}\right)^2} + a_3 e^{-\left(\frac{(x-b_3)}{c_3}\right)^2} + a_4 e^{-\left(\frac{(x-b_4)}{c_4}\right)^2} \quad (4.3)$$

with: $a_1 = 0.02336$; $b_1 = 112.3$; $c_1 = 4.02$; $a_2 = 0.01269$; $b_2 = 61.55$; $c_2 = 13.59$; $a_3 = 0.006351$; $b_3 = 38.92$; $c_3 = 30.59$; $a_4 = 0.002171$; $b_4 = 73.55$; $c_4 = 55.27$.

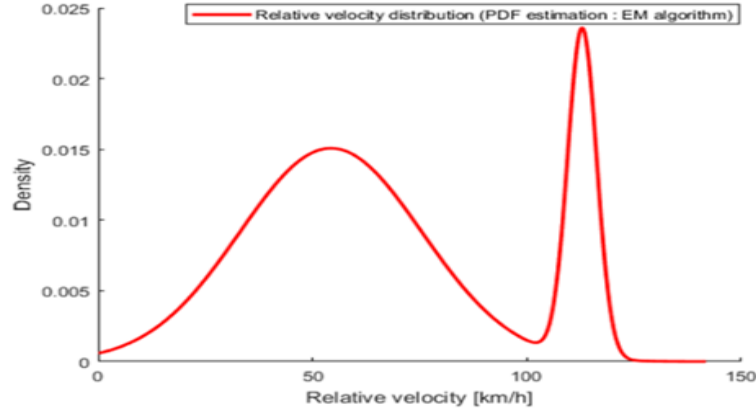


Figure 4-1 Relative velocity probability density

The sampling results from the relative velocity distribution is shown in Figure 4-1. To choose the optimal standard deviation for (σ) the proposal distribution, we apply a Kolmogorov-Smirnov test. For a more detailed explanation, we strongly suggest referring to our article entitled "New Field Operational Tests Sampling Strategy Based on Metropolis-Hastings Algorithm" [27].

We have two plots, the top one shows the true target distribution (in red), along with a histogram and its pdf distribution (in green) showing the distribution of samples obtained using the Metropolis sampler. The lower panel plots the actual Markov chain: the sequence of generated values. The chain shown in the lower panel moves rapidly across the whole distribution and covers the whole range without getting stuck in any one place; the acceptance rate here is 89%.

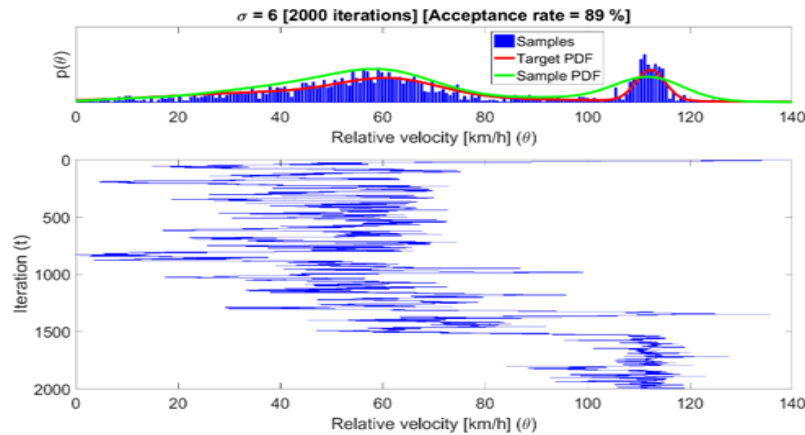


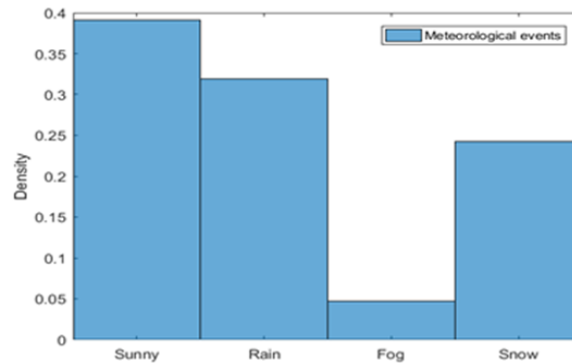
Figure 4-2 Sampling results ($\sigma = 6$): Relative Velocity [2000 iterations]

4.5 Modified Metropolis-Hastings algorithm for discrete probability distributions

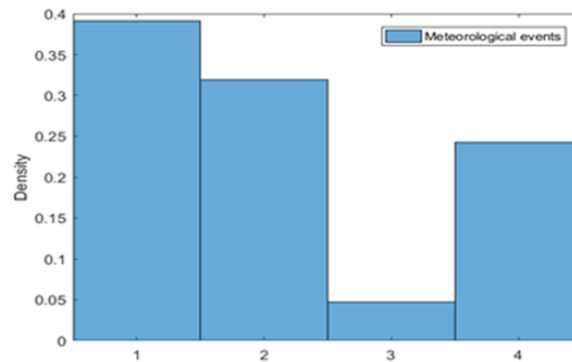
However, to sample from a discrete distribution $p(x)$ using the same algorithm; a modification to the algorithm is needed. For example, suppose we are interested in the distribution shown in Figure 4-3 (a). We have a distribution with four discrete events: sunny, rain, fog and snow (Also extracted from the SPMD database). The discrete probability distribution is given in Table 4-1:

Table 4-1 Discrete probability distribution

x	Sunny	Rain	Fog	Snow	$\sum p(X = x)$
$p(X = x)$	0.392	0.319	0.046	0.242	1



(a)



(b)

Figure 4-3 Discrete probability distribution: (a) events (b) events replaced with integers

4.5.1 Using multimodal distribution

For starters, we replace each event with an integer number (successive numbers) as shown in Figure 4-3 (b); in our example, we put: *Sunny* = 1, *Rain* = 2, *Fog* = 3, *Snow* = 4. Then, we replace this discrete distribution with a multimodal distribution (four modes since we have four events) centred at 1, 2, 3, 4 as shown in Figure 4-4. The estimated *PDF* is given by:

$$P(x) = a_1 e^{-\left(\frac{(x-b_1)}{c_1}\right)^2} + a_2 e^{-\left(\frac{(x-b_2)}{c_2}\right)^2} + a_3 e^{-\left(\frac{(x-b_3)}{c_3}\right)^2} + a_4 e^{-\left(\frac{(x-b_4)}{c_4}\right)^2} \quad (4.4)$$

With: $a_1 = 0.5127$; $b_1 = 1$; $c_1 = 0.4305$; $a_2 = 0.4184$; $b_2 = 2$; $c_2 = 0.4305$; $a_3 = 0.3179$; $b_3 = 4$; $c_3 = 0.4305$; $a_4 = 0.0613$; $b_4 = 3$; $c_4 = 0.4304$.

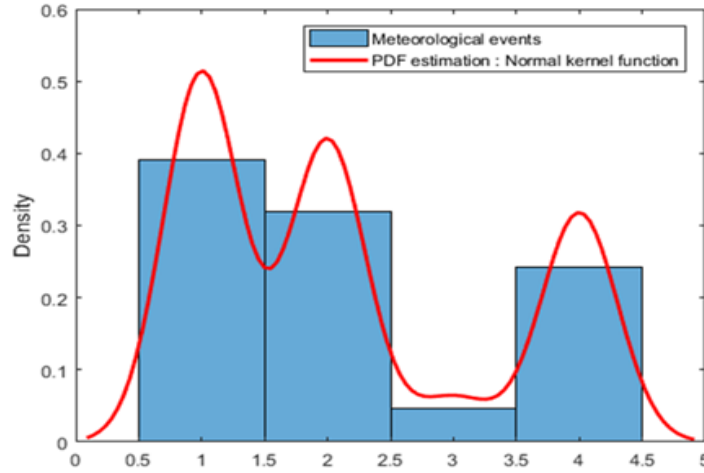


Figure 4-4 Estimated continuous probability density function of meteorological events

The next step consists of applying the Metropolis-Hastings algorithm to the continuous distribution, the sampling results shown in Figure 4-5.

Finally, a new step after the rejection and acceptance step is added to the modified algorithm. In this step, we assign every sample to its original class (event in this study). For this example, we proceed as follows:

- If: $0.5 \leq x^t \leq 1.5$ then $x_{or}^t = \text{sunny}$
- Else if: $1.5 < x^t \leq 2.5$ then $x_{or}^t = \text{rain}$

- Else if: $2,5 < x^t \leq 3,5$ then $x_{or}^t = \text{fog}$
- Else if: $3,5 < x^t \leq 4,5$ then $x_{or}^t = \text{snow}$
- Else: $x_{or}^t = 0$

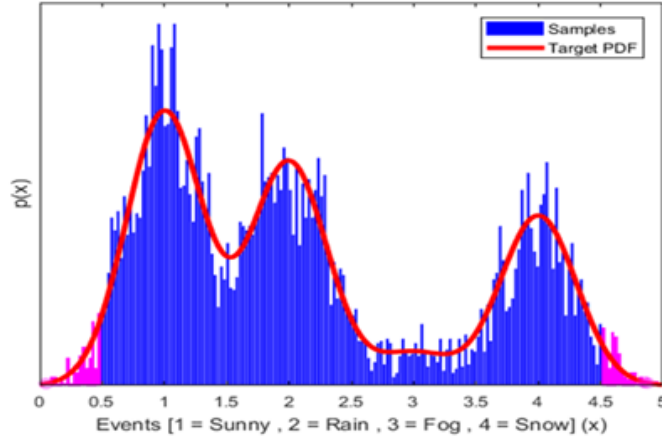


Figure 4-5 Sampling results of the continuous distribution

To verify the modified algorithm performance, a summary of the obtained results in comparison with the original distribution is given in Table 4-2. The new probability distribution of the retrieved samples using the modified sampler is shown in Figure 4-6. We notice an error of 3.4% (0.034). This is due the fact that some samples are taken into consideration despite of they are outside the region of interest (outside $]0.5, 4.5[$) as shown in magenta in Figure 4-5.

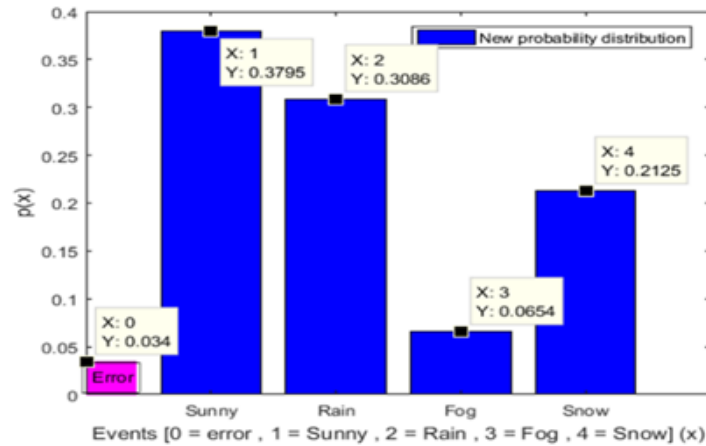


Figure 4-6 New probability distribution of the retrieved samples by the modified sampler based on a multimodal distribution

Table 4-2 Summary table: Original Distribution vs New Distribution (Multimodal distribution)

x	Sunny	Rain	Fog	Snow	$\sum p(X = x)$
$p(X = x)$	0.392	0.319	0.046	0.242	1
New $p(X = x)$	0.379	0.308	0.065	0.212	0.965
				Error	0.034

4.5.2 Using mixture of Generalized Gaussians

The main objective here is to reduce the error caused by the multimodal distribution's tail (as shown in Figure 4-5 and Figure 4-6). To do so, we replace the multimodal distribution with a mixture of Generalized Gaussians. Therefore, our discrete distribution (shown in Figure 4-3 (b)) will be replaced with a four Generalized Gaussians, one for each event, and the sum of these four distributions will be our new continuous distribution that we can sample from using the Metropolis-Hastings algorithm. The probability density function (PDF) of a Generalized Gaussian is defined by:

$$gg(x; \mu, \sigma, \rho) = \frac{1}{2\Gamma(1 + 1/\rho)A(\rho, \sigma)} \exp \left[-\left| \frac{x - \mu}{A(\rho, \sigma)} \right|^\rho \right], \quad x \in \mathbb{R} \quad (4.5)$$

Where $\mu \in \mathbb{R}$, $\rho, \sigma > 0$ and $A(\rho, \sigma) = \left[\frac{\sigma^2 \Gamma(1/\rho)}{\Gamma(3/\rho)} \right]^{1/2}$. $\Gamma(\cdot)$ is the Gamma function, μ is the mean, σ is the variance, the function $A(\rho, \sigma)$ is a scaling factor, and ρ is the shape parameter (see Figure 7). We can notice, when we can notice above, when $\rho = 1$, $gg(x; \mu, \sigma, \rho)$ is a Laplacian distribution, $\rho = 2$ corresponds to a Gaussian distribution. Whereas in the limiting cases $\rho \rightarrow +\infty$, the *pdf* in equation (5) converges to a uniform distribution in $(\mu - \sqrt{3}\sigma, \mu + \sqrt{3}\sigma)$ with an amplitude equal to $2\sqrt{3}\sigma$, and when $\rho \rightarrow +0$, the distribution becomes a degenerate one in $x = \mu$ [170].

A more complex density can be constructed from mixture of Generalized Gaussians. A mixture density, $p_M(x)$, is defined by:

$$p_M(x) = \sum_{j=1}^m \gamma_j p_j(x) \quad (4.6)$$

Where $p_j(x)$ are the constituent densities and γ_j are the associated probabilities with each constituent density.

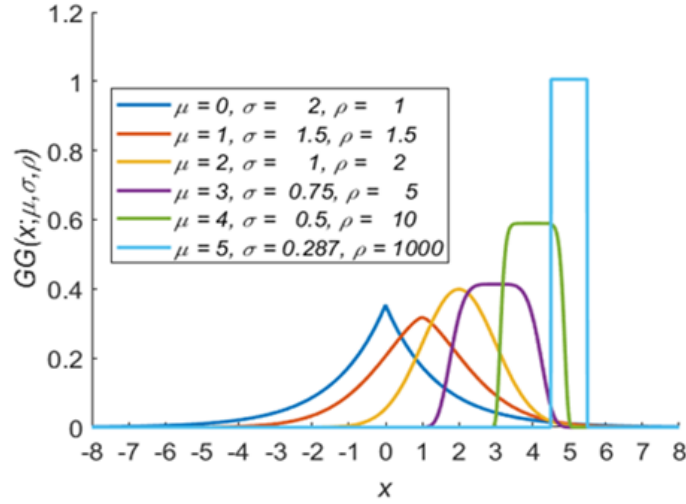


Figure 4-7 Generalized Gaussians with different locations, variance and shape parameters
(μ, σ, ρ)

We have four events that need to be modelled with a mixture of four Generalized Gaussians. The probability of each event is presented in table 2, so: $\gamma_1 = 0.392, \gamma_2 = 0.319, \gamma_3 = 0.046, \gamma_4 = 0.242$ and the needed mixture of Generalized Gaussian can be written as follows:

$$p_M(x) = 0.392 \times gg(x, \mu_1, \sigma_1, \rho_1) + 0.319 \times gg(x, \mu_2, \sigma_2, \rho_2) + 0.046 \times gg(x, \mu_3, \sigma_3, \rho_3) + 0.242 \times gg(x, \mu_4, \sigma_4, \rho_4) \quad (4.7)$$

As in the previous section, we replace each event with an integer number (successive numbers) as shown in Figure 3 (b); we put: *Sunny* = 1, *Rain* = 2, *Fog* = 3, *Snow* = 4. Then, we replace this discrete distribution with a mixture of Generalized Gaussians (four Generalized Gaussians since we have four events) centred at 1, 2, 3, 4 as shown in Figure 4-8. Therefore, we can directly deduce the values of the location parameters of each Generalized Gaussian, which is: $\mu_1 = 1, \mu_2 = 2, \mu_3 = 3, \mu_4 = 4$.

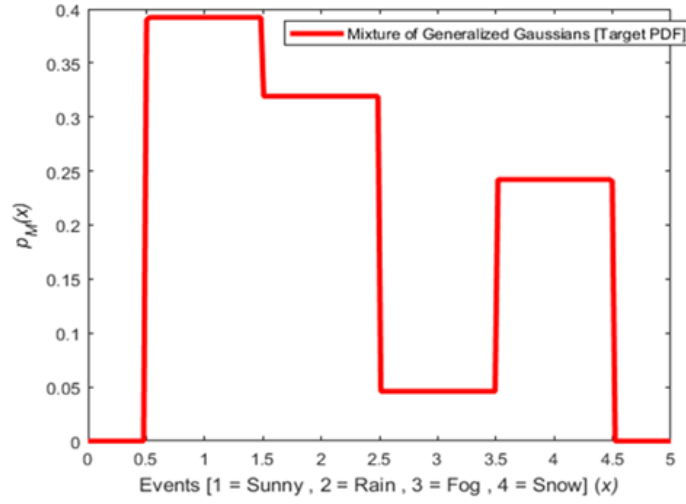


Figure 4-8 Continuous probability density distribution of discrete events (meteorological events) using a mixture of Generalized Gaussians (MGG)

To obtain a uniform distribution in $(\mu - 0.5, \mu + 0.5)$ with an amplitude of 1, the shape parameter needs to tend towards $+\infty$ and $2\sqrt{3}\sigma = 1$. We can calculate the variance value of each Generalized Gaussian using: $2\sqrt{3}\sigma = 1 \Rightarrow \sigma = 0.288$, so: $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 0.288$. We notice that $\rho = 1000$ gives us an almost perfect rectangular shape, as shown in Figure 4-7 (light blue), therefore, we limit ourselves to 1000. The shape parameter of each Generalized Gaussian is: $\rho_1 = \rho_2 = \rho_3 = \rho_4 = 1000$. The equation (5.7) becomes as follows in equation (5.8) and the mixture of Generalized Gaussians is shown in Figure 4-8:

$$p_M(x) = 0.392 \times gg(x, 1, 0.288, 1000) + 0.319 \times gg(x, 2, 0.288, 1000) + 0.046 \times gg(x, 3, 0.288, 1000) + 0.242 \times gg(x, 4, 0.288, 1000) \quad (4.8)$$

The next step consists of applying the Metropolis-Hastings algorithm to the MGG *pdf*, here is the sampling results shown in Figure 4-9.

As the previous section, a new step after the rejection and acceptance step is added to the modified algorithm. In this step, we assign every sample to its original class (event in this study). For this example, we proceed as follows:

- If: $0.5 \leq x^t \leq 1.5$ then $x_{or}^t = \text{sunny}$
- Else if: $1.5 < x^t \leq 2.5$ then $x_{or}^t = \text{rain}$

- Else if: $2,5 < x^t \leq 3,5$ then $x_{or}^t = \text{fog}$
- Else if: $3,5 < x^t \leq 4,5$ then $x_{or}^t = \text{snow}$
- Else: $x_{or}^t = 0$

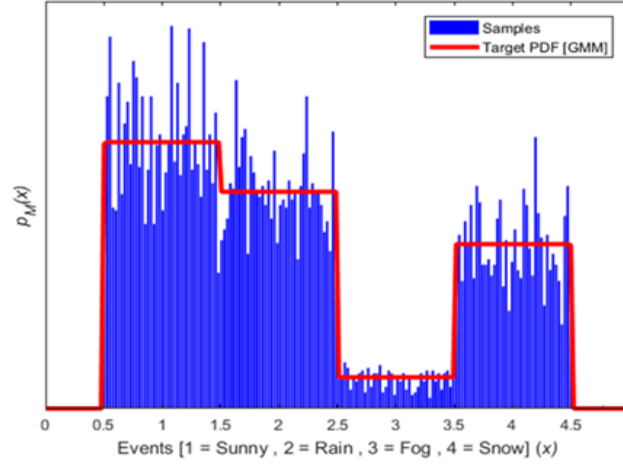


Figure 4-9 Sampling results of the continuous distribution (MGG)

To verify the modified algorithm performance, the new probability distribution of the retrieved samples using the modified sampler based on mixture of Generalized Gaussians is shown in Figure 4-10. A summary of the obtained results in comparison with the original distribution is given in Table 3. We notice that the error is practically zero (0.2%).

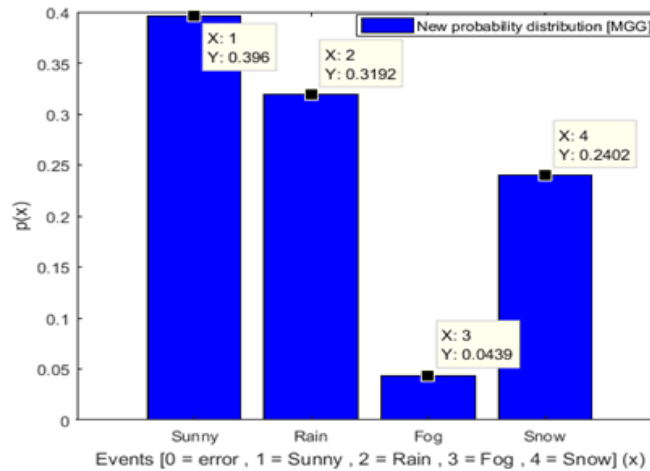


Figure 4-10 New probability distribution of the retrieved samples by the modified sampler based on a mixture of Generalized Gaussians

Table 4-3 Summary table: Original Distribution vs New Distribution (MGG)

x	Sunny	Rain	Fog	Snow	$\sum p(X = x)$
$p(X = x)$	0.392	0.319	0.046	0.242	1
MGG $p(X = x)$	0.369	0.319	0.043	0.240	0.998
				Error	0.002

4.6 Generalization: Modified Metropolis-Hastings Algorithm for Discrete Probability Distributions (MMH-DPD)

Suppose that our goal is to sample from a discrete probability distribution using the Modified Metropolis-Hastings algorithm. We define a discrete random variable X , which can take on any value from a finite or countably infinite set \mathcal{X} . We denote the probability of the event that $X = x$ by $p(X = x)$, or just $p(x)$. The listing of all possible values of X and their probability of occurring is called the probability distribution. This satisfies the properties $0 \leq p(x) \leq 1$ and $\sum_{x \in \mathcal{X}} p(x) = 1$. The state space \mathcal{X} is composed of n events: $\mathcal{X} = \{x_1, x_2, x_3, \dots, x_n\}$ and their respective probabilities are:

$$p(x_j) = \{p(x_1), p(x_2), p(x_3), \dots\} \quad (4.9)$$

First, the discrete distribution $p(x)$ will be replaced with n Generalized Gaussians, one for each event. The sum of these n distributions will be the new continuous distribution $p_M(x)$, Mixture of Generalized Gaussians, that we can sample from using the Metropolis-Hastings algorithm. The probability density function of a Generalized Gaussian is defined by:

$$gg(x; \mu, \sigma, \rho) = \frac{1}{2\Gamma(1+1/\rho)A(\rho, \sigma)} \exp \left[-\left| \frac{x-\mu}{A(\rho, \sigma)} \right|^\rho \right], \quad x \in \mathbb{R} \quad (4.10)$$

Where $\mu \in \mathbb{R}$, $\rho, \sigma > 0$ and $A(\rho, \sigma) = \left[\frac{\sigma^2 \Gamma(1/\rho)}{\Gamma(3/\rho)} \right]^{1/2}$. $\Gamma(\cdot)$ is the Gamma function, the parameter μ is the mean, σ is the variance, the function $A(\rho, \sigma)$ is a scaling factor, and ρ is the shape parameter. We can notice above, in the limiting cases $\rho \rightarrow +\infty$, the *pdf* in equation (10) converges to a uniform distribution in $(\mu - \sqrt{3}\sigma, \mu + \sqrt{3}\sigma)$ with an amplitude equal to $2\sqrt{3}\sigma$. A mixture density, $p_M(x)$, is defined by:

$$p_M(x) = \sum_{j=1}^m \gamma_j p_j(x) \quad (4.11)$$

Where $p_j(x)$ are the constituent densities and γ_j are the associated probabilities with each constituent density.

We have n events that need to be modelled with a mixture of n Generalized Gaussians. The probability of each event is presented in equation (8), so: $\gamma_1 = p(x_1), \gamma_2 = p(x_2), \gamma_3 = p(x_3), \dots, \gamma_n = p(x_n)$ and the needed mixture of Generalized Gaussian can be written as follows:

$$p_M(x) = p(x_1) \times gg(x, \mu_1, \sigma_1, \rho_1) + p(x_2) \times gg(x, \mu_2, \sigma_2, \rho_2) \\ + p(x_3) \times gg(x, \mu_3, \sigma_3, \rho_3) + \dots + p(x_n) \times gg(x, \mu_n, \sigma_n, \rho_n) \quad (4.12)$$

We replace each event with an integer number (successive numbers); we put:

$$x_1 = 1, x_2 = 2, x_3 = 3, \dots \quad (4.13)$$

Then, we replace the discrete distribution with a mixture of Generalized Gaussians (MGG), n Generalized Gaussians since we have n events, centred at $1, 2, 3, \dots, n$. Therefore, we can directly deduce the values of the location parameters of each Generalized Gaussian, which is: $\mu_1 = 1, \mu_2 = 2, \mu_3 = 3, \mu_4 = 4$.

To obtain a uniform distribution in $(\mu - 0.5, \mu + 0.5)$ with an amplitude of 1, the shape parameter needs to tend towards $+\infty$ and $2\sqrt{3}\sigma = 1$. We can calculate the variance value of each Generalized Gaussian using: $2\sqrt{3}\sigma = 1 \Rightarrow \sigma = 0.288$, so: $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 0.288$. We notice that $\rho = 1000$ gives us an almost perfect rectangular shape, as shown in Figure 7 (light blue), therefore, we limit ourselves to 1000. The shape parameter of each Generalized

Gaussian is: $\rho_1 = \rho_2 = \rho_3 = \rho_4 = 1000$. The equation (5.12) become as follows in equation (2.14):

$$p_M(x) = p(x_1) \times gg(x, 1, 0.288, 1000) + p(x_2) \times gg(x, 2, 0.288, 1000) \\ + p(x_3) \times gg(x, 3, 0.288, 1000) + \dots \times gg(x, n, 0.288, 1000) \quad (4.14)$$

The next step consists of applying the Metropolis-Hastings algorithm to the MGG ($p_M(x)$). The Metropolis sampler creates a Markov chain that produces a sequence of values:

$$x^{(1)} \rightarrow x^{(2)} \rightarrow \dots \rightarrow \dots \quad (4.15)$$

Where $x^{(t)}$ is the Markov chain state at iteration t . After the burn-in phase, the samples in the chain begin to reflect samples of the target distribution $p_M(x)$.

We initialize the algorithm with a random value $x^{(1)}$. We then use a proposition distribution $q(x|x(t-1))$ to generate a candidate x^* . There are a few minor technical constraints to be taken into consideration before choosing the proposition distribution, but for the most part, it can be anything that we like; which makes this algorithm a completely flexible method. A normal distribution centred on the current state is often used $q(x|x(t-1)) = N(x^{(t-1)}|\sigma^2)$; this is called the Random Walk Metropolis-Hastings algorithm (or the Metropolis Algorithm).

The next step is to accept or reject the proposal. The probability of accepting the proposal is as follows:

$$\alpha = \min \left(1, \frac{p_M(x^*)}{p_M(x^{(t-1)})} \right) \quad (4.16)$$

To decide on accepting or rejecting the proposal, we generate a uniformly distributed number μ between 0 and 1. If $\mu \leq \alpha$, we accept the proposition and the next state is equal to the proposition: $x^t = x^*$. If $\mu > \alpha$, we reject the proposition, and the next state is equal to the current state: $x^t = x^{(t-1)}$.

A new step after the rejection and acceptance step is added to the modified algorithm. In this step, we assign every sample to its original event. We proceed as follows:

- If: $0,5 \leq x^t \leq 1,5$ then $x_{or}^t = x_1$
- Else if: $1,5 < x^t \leq 2,5$ then $x_{or}^t = x_2$

- Else if: $2,5 < x^t \leq 3,5$ then $x_{or}^t = x_3$
- \vdots
- Else if: $3,5 < x^t \leq 4,5$ then $x_{or}^t = x_n$
- Else: $x_{or}^t = 0$

We continue to generate new proposals, accept or reject and assign every sample to its original event until the sampler reaches convergence. Here is a summary of the modified Metropolis-Hastings sampler (MMH-DPD) steps:

Modified Metropolis-Hastings Algorithm (MMH-DPD)

1. Replace each event with an integer number (successive numbers)

$$x_1 = 1, x_2 = 2, x_3 = 3, \dots$$

2. Create a new target

$$p_M(x) = p(x_1) \times gg(x, 1, 0.288, 1000) + p(x_2) \times gg(x, 2, 0.288, 1000) \\ + p(x_3) \times gg(x, 3, 0.288, 1000) + \dots \times gg(x, n, 0.288, 1000)$$

3. Set $t = 1$

4. Generate an initial value μ and x , and set $x^t = \mu$

5. Repeat

- a. $t = t + 1$

- b. Generate a proposition x^* from $N(x^{(t-1)} | \sigma^2)$

- c. Compute acceptance probability: $\alpha = \min\left(1, \frac{p(x^*)}{p(x^{(t-1)})}\right)$

- d. Generate μ from a uniform distribution: $Uniform(0,1)$

- e. If $\mu \leq \alpha$, accept the proposition and set $x^t = x^*$, else set $x^t = x^{(t-1)}$

- f. Assign every sample to its original event

- If: $0,5 \leq x^t \leq 1,5$ then $x_{or}^t = x_1$
- Else if: $1,5 < x^t \leq 2,5$ then $x_{or}^t = x_2$
- Else if: $2,5 < x^t \leq 3,5$ then $x_{or}^t = x_3$
- \vdots
- Else if: $(n-1),5 < x^t \leq n,5$ then $x_{or}^t = x_n$
- Else: $x_{or}^t = 0$

6. Until $t = T$
-

4.7 Conclusion

In this chapter, we proposed a novel modified Metropolis-Hastings algorithm (MMH-DPD) that can draw samples from discrete probability distributions. For starters, we replaced the discrete distribution with a multimodal distribution and the proposed algorithm worked as expected; however, a significant error (3.4%) compared to the original distribution was noticed. To reduce this error caused by the multimodal distribution's tail, we replace the multimodal distribution with a mixture of Generalized Gaussians. A new step after the rejection and acceptance step is added to the proposed algorithm where we assign every sample to its original event. The new probability distribution of the retrieved samples using the MMH-DPD algorithm based on a mixture of Generalized Gaussians has practically an error equal to zero (0.2%). We can reduce it to zero if we take larger number of iterations ($n \rightarrow \infty$). Thereby, we can conclude that the MMH-DPD sampler creates a Markov Chain that generate a sequence of x values denoted $(x^{(1)}, x^{(2)}, x^{(3)}, \dots)$, in such a way that as $n \rightarrow \infty$, we can guarantee that $x^{(n)} \sim p(x)$.

Chapitre 5

Proposal of a New Virtual Evaluation Approach of Preventive Safety Applications and Advanced Driver Assistance Functions - Application: AEB System -

5.1 Avant-propos

5.1.1 Auteurs et affiliation

Nacer Eddine Chelbi : Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Membre du Laboratoire sur l'Intelligence Véhiculaire (LIV).

Denis Gingras : Professeur, Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Responsable du Laboratoire sur l'Intelligence Véhiculaire (LIV).

Claude Sauvageau : Directeur essais et immobilisations, PMG Technologies inc. Blainville, Canada.

5.1.2 Date de soumission

26 janvier 2018.

5.1.3 État de l'acceptation

Version finale publiée.

5.1.4 Revue

IET Journals: IET Intelligent Transport Systems

5.1.5 Référence

N. E. Chelbi, D. Gingras, and C. Sauvageau, “Proposal of a new virtual evaluation approach of preventive safety applications and advanced driver assistance functions – application: AEB system,” IET Intelligent Transport Systems, vol. 12, no. 9, pp. 1148–1156, 2018.

5.1.6 Titre français

Proposition d'une nouvelle approche d'évaluation virtuelle des applications de sécurité préventive et de fonctions avancées d'aide à la conduite - Application: Système AEB -

5.1.7 Résumé français

Ce chapitre présente une proposition d'une nouvelle approche d'évaluation virtuelle d'applications de sécurité préventive et de fonctions avancées d'aide à la conduite. L'approche identifie les pires scénarios pour une fonction avancée d'aide à la conduite, système AEB dans cette étude, en se basant sur des tests opérationnels sur le terrain (FOT) [Safety Pilot Model Deployment (SPMD), dans cette étude]. Pour commencer, une description du système AEB étudié et une synthèse des scénarios de tests les plus pertinents sont présentées. Ensuite, une modélisation de la distribution de chaque paramètre de test extrait de la base de données SPMD en appliquant deux méthodes d'estimation (méthode du noyau et algorithme d'optimisation des attentes) est introduite. Une comparaison a été faite entre les deux méthodes pour choisir la meilleure. Ces distributions sont ensuite échantillonnées à l'aide de la stratégie d'échantillonnage proposée basée sur l'algorithme Metropolis-Hastings. Ensuite, l'idée est de prélever les échantillons de chaque paramètre récupéré avec cet échantillonneur, de les simuler sur un logiciel de simulation véhiculaire (PreScan) et d'obtenir les résultats de la simulation. Pour chaque test et en cas d'impact, un score proportionnel à la vitesse de réduction de l'impact est attribué. Enfin, une classification des risques est effectuée sur la base des résultats de la notation, ce qui permet de récupérer les cas à risque élevés et très élevés afin de créer un ensemble de scénarios les plus défavorables.

5.2 Abstract

This study presents a new virtual evaluation approach of preventive safety applications and advanced driver assistance functions. The approach identifies the worst-case scenarios for a given advanced driver assistance function, AEB system in this study, based on field operational tests (FOT) [Safety Pilot Model Deployment (SPMD), in this study]. The authors begin with a description of the studied AEB system and a synthesis of the most relevant test scenarios. Then, they model the distribution of each test parameter retrieved from the SPMD database by applying two estimation methods (kernel method and expectation-maximization algorithm). A comparison was made between the two methods to choose the best one. These distributions are then sampled using the proposed sampling strategy based on Metropolis-Hastings algorithm. Then, the idea is to take the samples of each parameter retrieved with this sampler, simulate them on a vehicular software simulator (PreScan) and to get their simulation results. For each test and in case of impact, a proportional score to the speed of impact reduction is attributed. Finally, a risk classification is done based on the scoring results which allows to recover high and very high-risk cases to build a set of worst-case scenarios.

5.3 Introduction

Safety analysis of a vehicular application is determined by its technical performance, its reliability, its interaction with the driver (human factors) and its level of road safety. Our study is based on the approaches developed and used by eIMPACT and PReVAL projects. These projects are part of ‘PReVENT Project’, one of the largest initiatives co-financed by the European Commission as an integrated road safety project. The initiative involved more than 50 European partners from leading vehicle manufacturers, equipment suppliers to the best research laboratories in vehicle intelligence. Since our work is essentially based on the evaluation part, we were more interested in the PReVAL project (Figure 5-1). The project proposes to extend the V-shaped cycle with evaluation-oriented steps [171]. One of the main objectives of PReVAL is the development of a harmonized framework for the evaluation of preventive safety applications and advanced driver assistance functions.

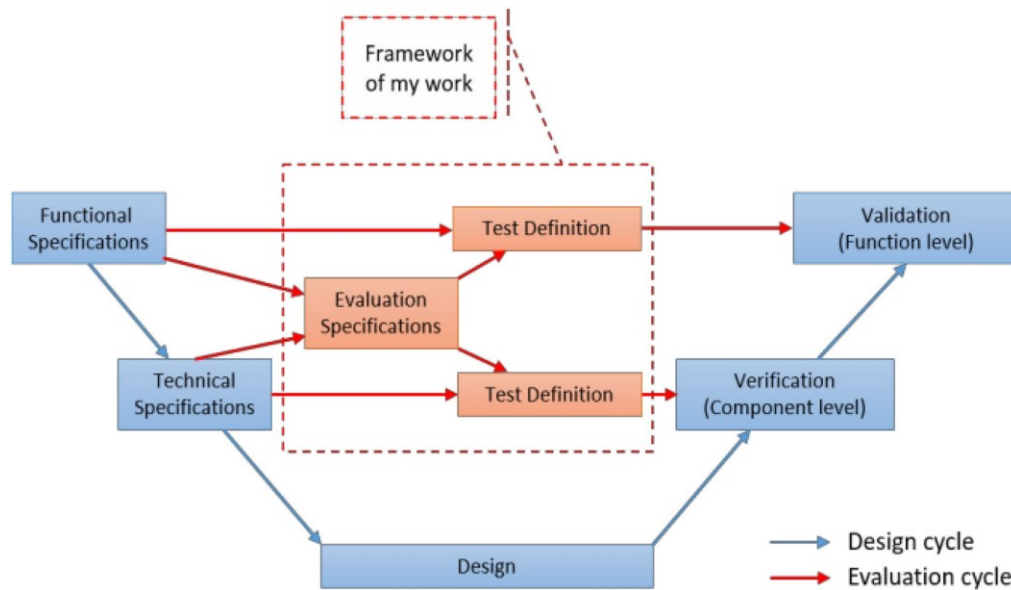


Figure 5-1 Adapted V-shaped design and evaluation cycle (PReVAL project) [171]

There are three main approaches for testing and validating advanced driving systems: field operational tests (FOT), tests on track and simulation. FOT are large scale testing programs designed to provide a more comprehensive assessment of the efficiency, quality, robustness and user acceptance of new vehicle technologies such as navigation, communication systems, advanced driver assistance systems (ADAS) and cooperative systems.

In Canada and the USA, manufacturers and importers must self- certify their vehicles to Motor Vehicle Safety Standards; CMVSS in Canada and FMVSS in the USA. In Europe, a third party performs a homologation of the vehicles that are manufactured or imported. New Car Assessment Program (NCAP) in the USA (US NCAP) as well as Euro NCAP in Europe publishes test procedures to rate the performance of vehicles in order to help consumers choose their vehicles. Certification and NCAP testing can be performed in test tracks in specialized test centres such as Transport Canada's Motor Vehicle Test and Research Centre, managed by PMG Technologies.

Real driving tests are proven to be the most reliable solution, but unluckily, it is not the optimal one: due to time and cost issues, these tests cannot cover all the operating conditions. Moreover, test scenarios cannot be reproduced exactly as they first happened. One way to bypass this problem is by using virtual validation tools; vehicular simulators make a very good example.

On the other hand, using simulations alone for the validation procedure is often insufficient because of a large number of virtual tests that should be done.

As stated by National Highway Traffic Safety Administration (NHTSA), to demonstrate the expected performance of a highly automated vehicles system, test approaches should include a combination of simulation, test track and on road testing [25]. Therefore, in this study, we propose a new virtual validation approach, shown in Fig. 2, of preventive safety applications and advanced driver assistance functions, which involves FOT, a probabilistic approach and simulation tests using a vehicular simulation software (PreScan [65]). This approach depends on our proposed sampling strategy which is based on Markov chain Monte Carlo (MCMC) methods that use Metropolis-Hastings algorithm to generate samples from probability distributions of FOT; the safety pilot model deployment (SPMD) in our case.

The main contribution of our work is the development of a validation approach of preventive safety applications and advanced driver assistance functions, which combines the three validation approaches: on road, test track and simulation (see Figure 5-2). To the best of our knowledge, such combination has not been studied in the literature.

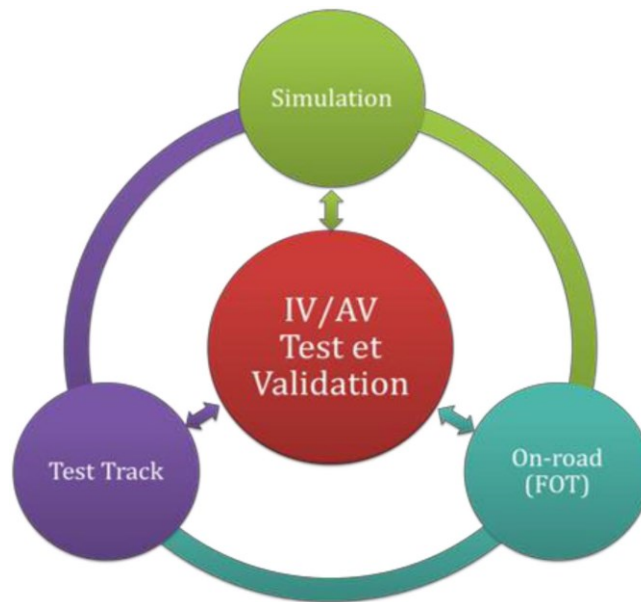


Figure 5-2 Combination of the three validation approaches

5.4 Proposed evaluation approach

For better visualization, here is the proposed evaluation approach architecture in Figure 5-3. Our approach is broken down into four parts. In this study, we will discuss three parts. A first part (in orange), which includes a description of the studied autonomous emergency braking (AEB) system and of its different evaluation criteria to propose a synthesis of the most relevant test scenarios with the operating parameters of each scenario. A second part (in blue) that deals with the FOT database (SPMD), the implementation of a new sampling strategy (more detailed in our other article [27]) based on Metropolis-Hastings algorithm and at the end of this part, the implementation of virtual tests on a vehicular software simulator (PreScan). The third part (in green) focuses on collecting of simulation results, then assess the risk of each test, to classify them, and this allows us to recuperate high risk and very high-risk cases to build a set of worst-case scenarios. The rest of the fourth part will be discussed in more detail in a future work.

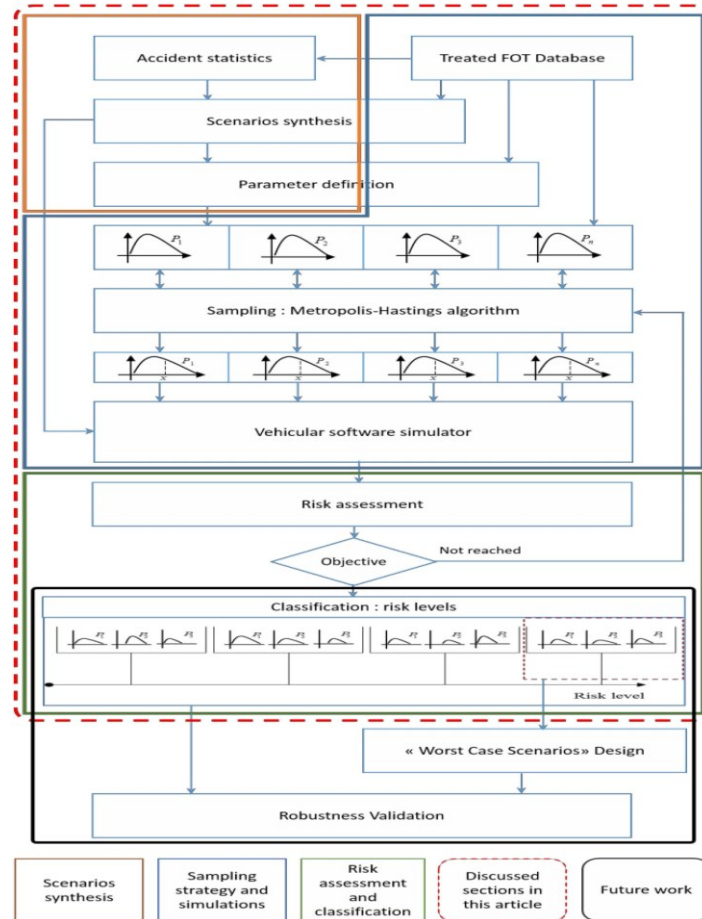


Figure 5-3 Proposed evaluation approach

5.5 Scenarios synthesis

5.5.1 AEB system description

The studied AEB system is composed of two radar sensors: a long-range radar (LRR) with a range of 150 m and a short-range radar (SRR) with a range of 30 m but with a wider beam. Use cases description of an AEB system may include situational descriptors such as target type, road type, environmental conditions, traffic conditions and so on:

1. Target type: vehicle, pedestrian;
2. Environmental conditions:
 1. Weather: sunny, rain, snow, fog;
 2. Brightness: 0.5–100,000 lux;
 3. Road conditions: dry, wet.
3. Traffic conditions:
 1. Speed: 0.5–120 km/h.

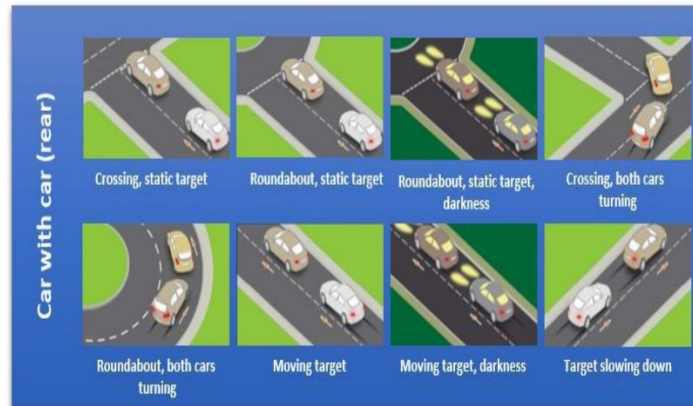
The system begins to warn the driver 2.6 s before the expected impact [time to collision (TTC)] = 2.6 s). At $TTC = 1.6$ s, a braking pressure of 40% is applied. This gives extra time for the driver to react and increases the possibility of breaking or avoiding the collision. If the driver starts braking, the system will help him with extra strength, reaching the maximum braking value. If no reaction is taken, at $TTC = 0.6$ s, a full brake pressure is applied [65].

5.5.2 Accidents classification

According to Thatcham study based on STATS 19 and OTS statistics [172], we provide in Figure 5-4 a summary of most common types of accidents that can be mitigated or avoided using an AEB systems.

5.5.3 Evaluation indicators

Indicators are generally defined to measure the efficiency of the system. To do so, these indicators must be able to clearly reflect the behaviour of the system and the performance related to its operation.



a



b

Figure 5-4 Types of accidents (a) Vehicle with vehicle, (b) Vehicle with pedestrian

5.5.3.1 Performance indicators

To analyze the quality interventions made by the system, we need indicators that are related particularly to intervention systems. For the AEB systems we can use distance to target after braking (in case of successful avoidance), the speed of impact and its reduction rate (in case of unsuccessful avoidance).

5.5.3.2 Decision Indicators

This section contains indicators related to parameters that are used for function activations. They are used to find out if the system is working properly when needed. For AEB systems, three indicators are generally used: TTC, distance to the target and relative speed between the vehicle

and the detected target. In our case, these indicators (performance and decision indicators) will be used to classify different tests according to their risk levels.

5.5.4 Most Representative Factors of AEB Scenarios

For AEB performance evaluation, test scenarios should not represent only collision cases that are frequently encountered in its life, but also the most significant physical situations. Therefore, for performance evaluation of an AEB system in these various situations, many factors should be considered such as driving modes, road types, environment and elements of the AEB system (see Figure 5-5) [173], [174]. The studied AEB system is composed of two radar sensors: a LRR with a range of 150 m and a SRR with a range of 30 m but with a wider beam. Use cases description of an AEB system may include situational descriptors such as target type, road type, environmental conditions, traffic conditions and so on.

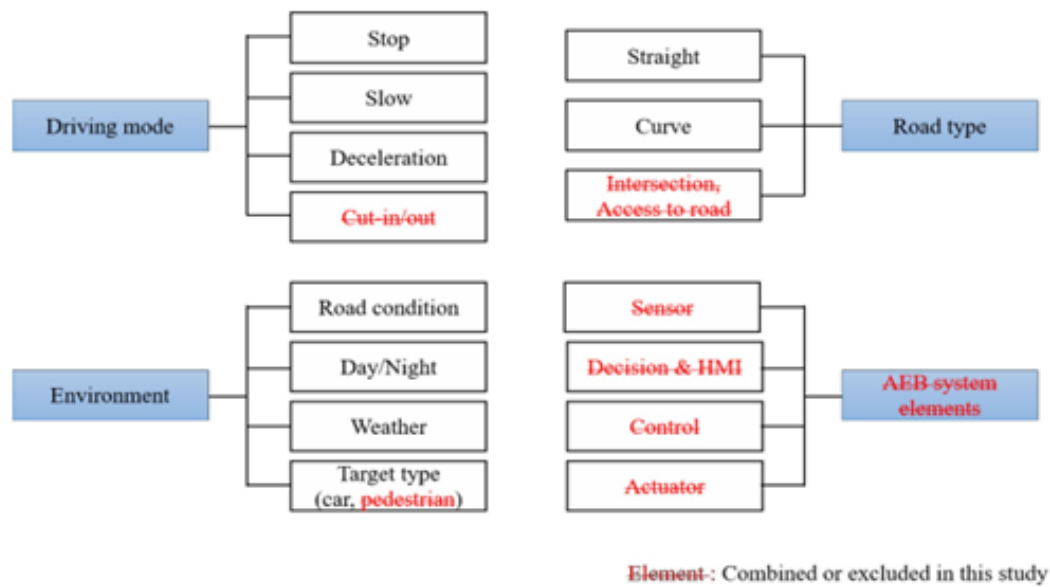


Figure 5-5 Most representative factors of AEB scenarios

5.5.5 Proposed test scenarios (synthesis)

Based on the above-mentioned factors and the existing AEB test protocols (NHTSA, Euro-NCAP) [102], [173], [174], a synthesis of test scenarios is proposed in Table 5-1. This synthesis represents a starting point and an inventory of the most relevant scenarios that will be used throughout our work. Driving mode ‘cut-in/out’ (combination of two driving modes ‘slow and deceleration’), road type ‘Intersection’, target type ‘pedestrian’ and elements of the AEB system

(black-box approach) were excluded in this study. The same approach with the appropriate changes will be applied to the pedestrian as a target. We excluded the pedestrian as a target to avoid overloading this study.

Table 5-1 Scenarios synthesis

Designation	Scenarios	Target	Variables
AEBJ1-1	Straight line - day	Stopped	
AEBJ1-2	Straight line - day	Slow	
AEBJ1-3	Straight line - day	Decelerating	
AEBN1-1	Straight line -night	Stopped	
AEBN1-2	Straight line -night	Slow	Distance: Dst
AEBN1-3	Straight line -night	Decelerating	Speed: V_s, V_t
AEBJ2-1	Curve - day	Stopped	Deceleration: A_t
AEBJ2-2	Curve - day	Slow	Weather: Sunny,
AEBJ2-3	Curve - day	Decelerating	rain, snow, fog
AEBN2-1	Curve - night	Stopped	
AEBN2-2	Curve - night	Slow	
AEBN2-3	Curve - night	Decelerating	

5.6 Sampling strategy and simulations

This study is based on our proposed sampling strategy in [27]. An appropriate sampling strategy is based on MCMC methods, using Metropolis-Hastings algorithm to generate samples from probability distributions of FOT; the SPMD in our case. For each step of the sampling strategy, we will give a brief explanation in the following subsections. For a more detailed explanation, we strongly suggest referring to our article entitled “New Field Operational Tests Sampling Strategy Based on Metropolis- Hastings Algorithm” [27].

5.6.1 SPMD database

We opted for the “SPMD” program, a research initiative highlighting the application of connected vehicle safety technologies and ‘V2V, V2I (DSRC)’ systems in real driving conditions. This program recorded 2842 equipped vehicles in Ann Arbor, Michigan for more

than 2 years. In April 2015, 34.9 million miles were recorded, making it one of the largest databases made public [17].

5.6.2 Meteorological data

Since the SPMD database contains only 2 months of meteorological data, we opted for Weather Underground website (www.wunderground.com). A site that gives access to weather conditions from all over the world. The data used in this study come from Montreal-Mirabel International Airport (CYMX) weather station from 1 January 2007 to 31 December 2016. In our case, we assume that the weather conditions will have a more significant influence on the vehicle behaviour (slippage, etc.) than on the LRR and SRR radars used by the AEB system.

5.6.3 Data Extraction

In this study, we will apply our approach to the first scenario: AEBJ1-1, a scenario with a two-vehicle configuration, as shown in Figure 5-6, a standstill target vehicle and a test vehicle that approaches at different speeds and distances. The data needed to be extracted are: relative distance, relative speed and weather conditions. In order to extract only related data to the same conditions of this first scenario, two search conditions were applied: type of target needs to be a vehicle with a speed equal to zero.

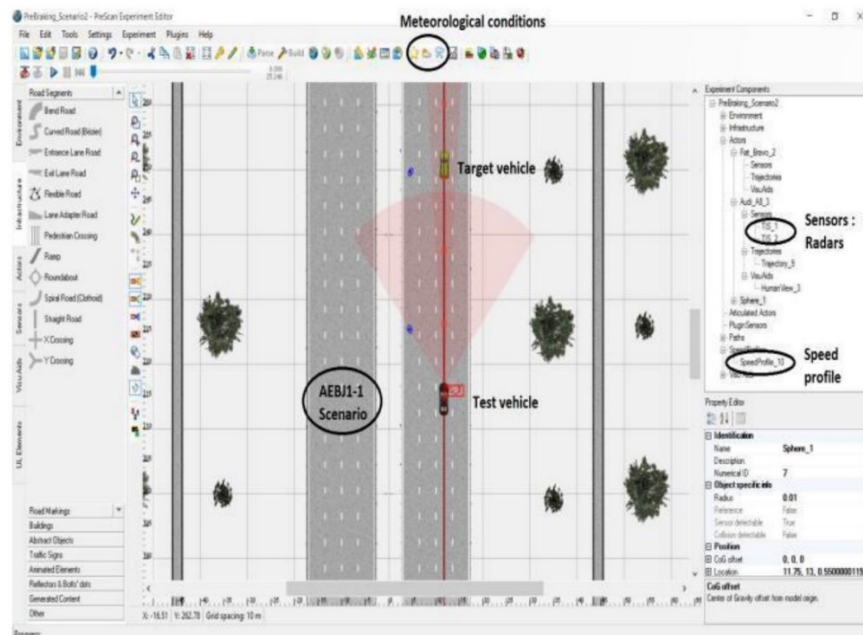


Figure 5-6 PreScan GUI of the first scenario AEBJ1-1

5.6.4 Probability density function (PDF) estimation

To estimate the PDF model of each parameter, we applied two estimation methods: a kernel density estimation (KDE) and an estimation based on the expectation-maximisation (EM) algorithm.

5.6.4.1 Kernel density estimation

Kernel density estimation is a simple non-parametric method for estimating the probability density of a random variable. It makes it possible to estimate the density at any point based on a sample from a statistical population. This is a generalization of the histogram estimation method [175].

Let $x_1, x_2, \dots, x_N \sim f$ an independent and identically distributed sample (i.i.d) drawn from a certain distribution with unknown density f . We are interested in estimating the form of this function f . Then the nonparametric estimator by the kernel density estimation method is given by:

$$\hat{f}_h(x) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{x - x_i}{h}\right) \quad (5.1)$$

where K is a kernel and h is a parameter called window. In our case, K is chosen as a standard normal distribution ($\mu = 0$ and $\sigma = 1$):

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \quad (5.2)$$

5.6.4.2 Gaussian mixture model estimation using EM algorithm

A Gaussian Mixture Model (GMM) is used to model the parametric estimation of a distribution of a random variable as a sum of several Gaussians. It is then necessary to determine the mixing probability, the mean and the variance of each Gaussian. This is an iterative technique and it is done most often via the Expectation-Maximization (EM) algorithm. The EM algorithm is composed of two steps: expectation (E step) and maximization (M step) [176].

Expectation (E step): In this step, we calculate the probability that each data point belongs to each cluster. To begin, we will need the PDF equation of a multivariate normal distribution.

$$g_i(x) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} e^{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)} \quad (5.3)$$

where $g_j(x)$ is a multivariate normal distribution PDF of the cluster j , j is the cluster number, x is the input vector, n is the length of the input vector, Σ_j is the covariance matrix $n \times n$ for cluster j , $|\Sigma_j|$ is the determinant of the covariance matrix, Σ_j^{-1} is the inverse of the covariance matrix.

The probability that a point i is part of the cluster j can be calculated by:

$$w_j^i = \frac{g_i(x)\Phi_j}{\sum_{l=1}^k g_l(x)\Phi_l} \quad (5.4)$$

where w_j^i is the probability that a point i is part of the cluster j , $g_j(x)$ is the multivariate normal distribution PDF of the cluster j , Φ_j is the prior probability of the cluster j , k is the cluster number. We apply this equation to every data point and cluster.

Maximization (M step): The update equations of the maximization step are given below:

$$\Phi_j := \frac{1}{n} \sum_{i=1}^n w_j^{(i)} \quad (5.5)$$

$$\mu_j := \frac{\sum_{i=1}^n w_j^{(i)} x^{(i)}}{\sum_{i=1}^n w_j^{(i)}} \quad (5.6)$$

$$\Sigma_j := \frac{\sum_{i=1}^n w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^n w_j^{(i)}} \quad (5.7)$$

The prior probability of a cluster j , denoted Φ , is computed as the average of probability that a data point belongs to a cluster j . The μ equation of the cluster j is the average of all data points in the learning data set, each point being weighted by its cluster belonging probability. We repeat the same iterations until convergence.

5.6.4.3 Kernel density estimation vs EM algorithm estimation

After plotting the histograms of each parameter (all figures in this subsection can be found in appendix section 5.12.1), we noticed that some parameters have unimodal distributions (relative distance (Figure 5-12), humidity (Figure 5-15) and other parameters have multimodal distributions (relative velocity (Figure 5-13), temperature (Figure 5-14), visibility (Figure 5-16). For unimodal distributions and for simplicity reasons, we have chosen to use the kernel density estimation. For multimodal distributions, a comparison was made between the two estimation methods to choose the best one. We noticed that for the studied cases, the kernel density estimation method offers a better estimate than the EM algorithm (see Figure 5-13, Figure 5-14, Figure 5-16), so we have chosen the kernel density estimation results. The estimation results can be found in the appendix section 5.12.1.

5.7 Metropolis-Hastings Algorithm Implementation

Our goal is to sample from a target probability density, we begin with the relative velocity $p(\theta)$ presented by equation (6.13) in appendix 5.12.1.2, with $0 < \theta < 140$ km/h. The Metropolis sampler creates a Markov chain that produces a sequence of values [168]:

$$\theta^{(1)} \rightarrow \theta^{(2)} \rightarrow \dots \dots \dots \quad (5.8)$$

where $\theta^{(t)}$ is the Markov chain state at iteration t . After the burn-in phase, the samples in the chain begin to reflect samples of the target distribution $p(\theta)$.

We initialize the algorithm with a random value $\theta^{(1)}$ between 0 and 140. We then use a proposition distribution $q(\theta|\theta(t-1))$ to generate a candidate θ^* . There are a few minor technical constraints to be taken into consideration before choosing the proposition distribution, but for the most part, it can be anything that we like; which makes this algorithm a completely flexible method. A normal distribution centred on the current state is often used $q(\theta|\theta(t-1)) = N(\theta^{(t-1)}|\sigma^2)$; this is called the Random Walk Metropolis-Hastings algorithm (or the Metropolis Algorithm), and this is what we are going to apply in this article.

The next step is to accept or reject the proposal. The probability of accepting the proposal is as follows:

$$\alpha = \min\left(1, \frac{p(\theta^*)}{p(\theta^{(t-1)})}\right) \quad (5.9)$$

To decide on accepting or rejecting the proposal, we generate a uniformly distributed number μ between 0 and 1. If $\mu \leq \alpha$, we accept the proposition and the next state is equal to the proposition: $\theta^t = \theta^*$. If $\mu > \alpha$, we reject the proposition, and the next state is equal to the current state: $\theta^t = \theta^{(t-1)}$. We continue to generate new proposals and accept or reject until the sampler reaches convergence. Here is a summary of the Metropolis sampler steps:

Metropolis-Hastings algorithm	
1.	Set $t = 1$
2.	Generate an initial value μ and θ , and set $\theta^t = \mu$
3.	Repeat
1.	$t = t + 1$
2.	Generate a proposition θ^* from $N(\theta^{(t-1)} \sigma^2)$
3.	Compute acceptance probability: $\alpha = \min\left(1, \frac{p(\theta^*)}{p(\theta^{(t-1)})}\right)$
4.	Generate μ from a uniform distribution: $Uniform(0,1)$
5.	If $\mu \leq \alpha$, accept the proposition and set $\theta^t = \theta^*$, else set $\theta^t = \theta^{(t-1)}$
4.	Until $t = T$

The sampling results can be found in appendix section (5.12). And here is a summary diagram of the algorithm:

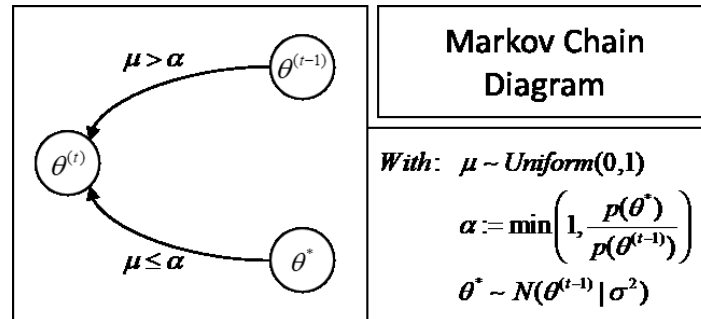


Figure 5-7 Markov Chain diagram

5.8 Scenario implementation (PreScan Simulator)

Before proceeding to the simulation steps, here is the GUI definition of the first scenario AEBJ1-1 in Figure 5-6. Two radar sensors (LRR, SRR) were added to the test vehicle. Speed profile and weather conditions are defined before each simulation.

5.9 Simulation Steps and Results

To apply and confirm the concept of the proposed evaluation approach, 100 virtual tests were performed. To prove our concept, in this study we took 100 samples of each parameter manually (relative distance, relative speed, temperature, humidity, visibility, events) retrieved with the Metropolis-Hastings sampler; supply them to the simulator one by one, simulate the scenario, retrieve the results (TTC, deceleration, impact speed, distance), and finally, calculate the impact speed reduction rate using:

$$T_r = \frac{V_{vehicle} - V_{impact}}{V_{vehicle}} \times 100\% \quad (5.10)$$

In Table 5-2, we give some examples of data that was provided to the simulator. Table 5-3 shows some examples of data recovered and calculated after each simulation. TTC is the collision time. To tell the difference if a collision was avoided or not, TTC takes a positive sign (time before the warning is triggered) in the first case and a negative sign (warning activation time) in the second case.

Table 5-2 Examples of samples provided to simulator

Distance (m)	Speed (km/h)	Temperature (K)	Humidity (%)	Visibility (km)	Events
34	60	266.15	93	11	snow
9	57.3	266.15	100	28	clear
34	50.9	269.15	100	12	rain
116	58.7	267.15	100	4	snow
32	64.9	272.15	100	20	snow

Table 5-3 Examples of data recovered and calculated after each simulation

TTC (s)	Deceleration (km/h²)	Impact speed (km/h)	Distance (m)	Collision (yes/no)	Reduction rate (%)
-2.47	-9.9	12	0	yes	80
-0.38	-9.94	54	0	yes	21
0.05	-8.86	0	1.21	no	100
-3.42	-9.4	3.5	0	yes	94
-2.09	-9.92	20.1	0	yes	69

In this study, we did not have any irrational combination. However, to make sure of that, we added a filtering condition (combination between temperature and the event). We found the highest temperature (T_{maxsn}) recorded during a snowy day (in the last 10 years in Montreal), if we find a combination where it snows and exceeds T_{maxsn} at the same time, the combination is rejected.

In the next step (in future work), we will automate our testing approach to run all the samples that have been extracted by the Metropolis-Hastings algorithm. Then we apply the same process to the rest of the scenarios mentioned in Table 5-1.

5.10 Risk assessment and classification

5.10.1 Risk assessment

To assess the risk of each test, we will apply the following scoring system: in case of an impact, a proportional score to the speed of impact reduction is attributed. This score calculation is given by the following equation:

$$Note^{(t)} = \left(\frac{v_{test(t)} - v_{impact(t)}}{v_{test(t)}} \right) \times point_{test(t)} \quad (5.11)$$

where $point_{test}$ is the number of points available for different speeds. The score is deduced from the scoring table in Figure 5-8, where green = 1.000; yellow = 0.750; orange = 0.500; brown = 0.250; red = 0.000.

The scoring table was inspired by the Euroncap AEB systems TEST PROTOCOL [102]. In our case, we extrapolated the colour scheme from the scoring table of the AEB City to higher test speeds (>80 km/h). The scoring standard is a function of three factors: vehicle speed, impact speed and points of the test. The calculation of the later (points of the test) is based on the frequency of accidents, where the most frequent accidents are more weighted than others. Therefore, when we have other scenarios (curved driving, etc.), the weights of test points will change accordingly to the frequency of accidents in each specific scenario.

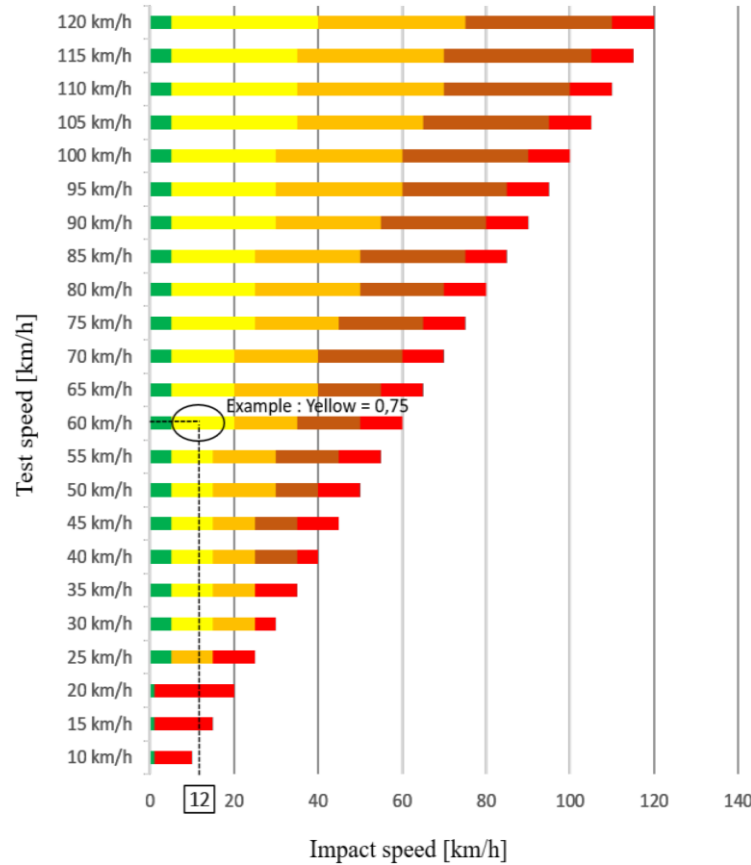


Figure 5-8 Scoring table

We take as an example the first row of Table 5-2, with a speed of 60 km/h and impact speed of 12 km/h from the first row of Table 5-3, with this information we can retrieve the number of points awarded from the scoring table in Figure 5-8, we calculate its score $Note^{(1)}$ and we find 0.6.

$$Note^{(1)} = \left(\frac{v_{test(1)} - v_{impact(1)}}{v_{test(1)}} \right) \times point_{test(1)} = \left(\frac{60 - 12}{60} \right) \times 0.75 = 0.6 \quad (5.12)$$

Therefore, the assigned score for this first test is 0.6. We repeat the same approach with the rest of the tests; Table 5-4 shows the scoring results of some tests.

Table 5-4 Scoring results of a few tests

Distance (m)	Speed (km/h)	Impact speed (km/h)	Reduction rate (%)	Point	Note	1/Distance
34	60	12	80	0.75	0.6	0.029
9	57.3	45	21	0.25	0.054	0.110
116	58.7	3.5	94	1	0.941	0.009
32	64.9	20.1	69	0.75	0.518	0.031

5.10.2 Risk classification

In this study, a visual labelling is applied. We used the basic idea of *K*-Means clustering. We tried to split our data into *k* groups based on their scatter plot. By analysing the scoring histogram (Figure 5-9), we found that the data could be grouped into four groups. Therefore, a classification of four risk levels can be proposed:

1. Very high risk: $0.9 < 1 - \text{Note}(t) \leq 1$
2. High risk: $0.8 < 1 - \text{Note}(t) \leq 0.9$
3. Low risk: $0.5 < 1 - \text{Note}(t) \leq 0.8$
4. Very low risk: $0 < 1 - \text{Note}(t) \leq 0.5$

The idea consists of recovering high and very high-risk cases to build a set of worst-case scenarios (see Figure 5-9). We put: $1 - \text{Note} = \text{Score}$.

By analysing the other parameters, we can notice that the scores assigned to each test are inversely proportional to the inverse of the distance (Table 5-4). We can, therefore, add a second labelling compared to the inverse of the distance, which allows us to propose three levels of distance (Figure 5-10).

1. Far: $0 < (1 / \text{Distance}) \leq 0.06$
2. Moderately near: $0.06 < (1 / \text{Distance}) \leq 0.13$

3. Near $0.13 < (1/\text{Distance}) \leq 0.2$

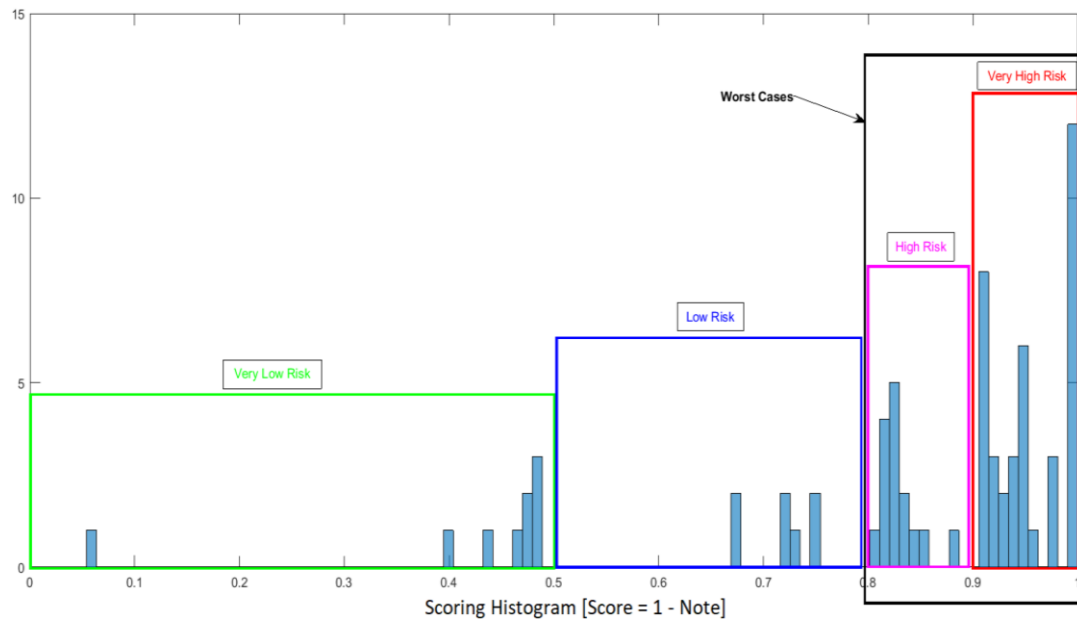


Figure 5-9 Scoring histogram (score = 1–Note)

We plotted the inverse of the distance as a function of score in Figure 5-10 and Figure 5-11. We found that the scattered data could be grouped into 9–12 groups. This new labelling allows us to have a better understanding and a better classification of the tests. To have a better separation between the risk levels, we have chosen to create 12 risk levels, with ‘1’ being the lowest and ‘12’ being the highest (Figure 5-11).

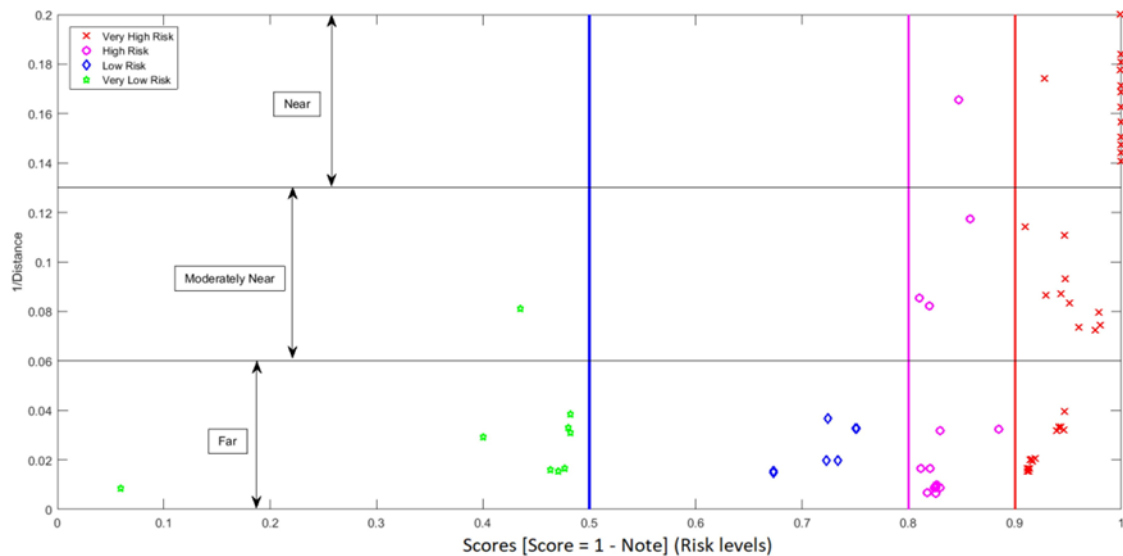


Figure 5-10 1/Distance as a function of scores (risk levels) (score = 1–Note)

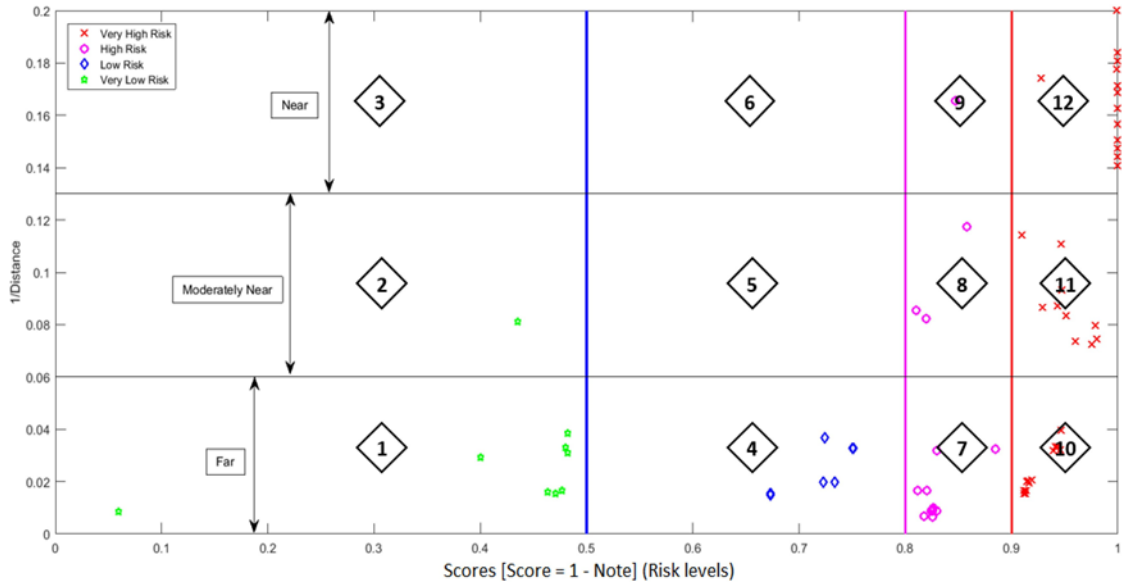


Figure 5-11 Risk classification: 12 levels (score = 1–Note)

5.11 Conclusion

This In this study, we proposed a new virtual evaluation approach of preventive safety applications and advanced driver assistance functions based on Metropolis-Hastings algorithm. The goal was to identify the worst-case scenarios for a given ADAS application (AEB system), based on FOT database analysis. With 100 virtual tests, we were able to demonstrate how to extract the worst cases based on a risk classification. The application of Metropolis-Hastings algorithm allowed us to propose a sampling strategy, that is:

1. Easy: Easy implementation of the algorithm
2. Robust: Almost a guarantee of convergence towards the exact distribution
3. A global approach: Weighted coverage over the entire operating space.

The quality of our results depends mainly on the applied sampling strategy quality; which itself depends on the quality of the PDF estimation and the optimization of Metropolis-Hastings algorithm. In future work, we will consider a more advanced parameters optimization concepts for the Metropolis-Hastings algorithm, such as initialization, number of iterations and so on. For risk classification, we will focus on applying more advanced classification techniques (supervised or unsupervised techniques).

5.12 Appendix

5.12.1 PDF model's estimation results

5.12.1.1 Relative distance D_{st}

$$P(x) = a_1 e^{-\left(\frac{(x-b_1)}{c_1}\right)^2} + a_2 e^{-\left(\frac{(x-b_2)}{c_2}\right)^2} + a_3 e^{-\left(\frac{(x-b_3)}{c_3}\right)^2} + a_4 e^{-\left(\frac{(x-b_4)}{c_4}\right)^2} \quad (5.13)$$

where $a_1 = 0.01806$; $b_1 = 12.13$; $c_1 = 5.022$; $a_2 = 0.01127$; $b_2 = 19.92$; $c_2 = 9.231$; $a_3 = 0.008861$; $b_3 = 36.54$; $c_3 = 20.24$; $a_4 = 0.002773$; $b_4 = 86.48$; $c_4 = 72.42$.

Figure 5-12 shows the relative distance and its PDF estimate using KDE method.

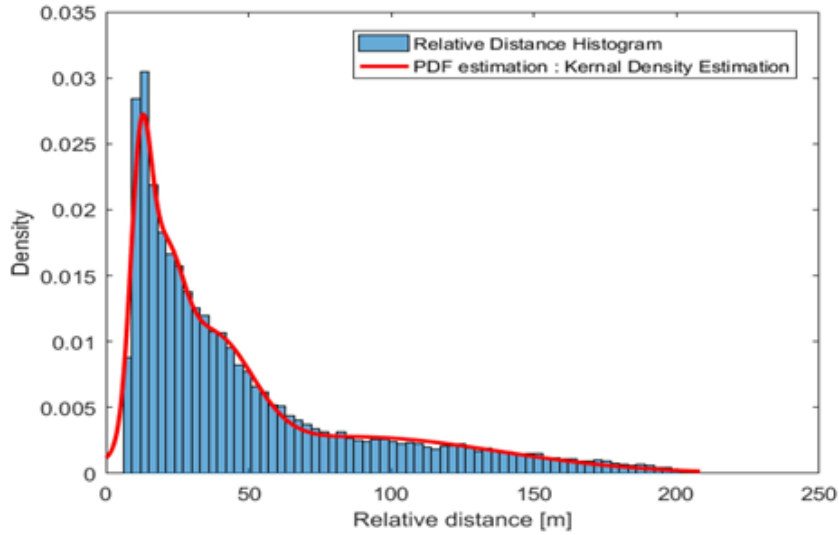


Figure 5-12 PDF estimation: Relative Distance D_{st}

5.12.1.2 Relative velocity V_s

$$P(x) = a_1 e^{-\left(\frac{(x-b_1)}{c_1}\right)^2} + a_2 e^{-\left(\frac{(x-b_2)}{c_2}\right)^2} + a_3 e^{-\left(\frac{(x-b_3)}{c_3}\right)^2} + a_4 e^{-\left(\frac{(x-b_4)}{c_4}\right)^2} \quad (5.14)$$

where $a_1 = 0.02336$; $b_1 = 112.3$; $c_1 = 4.02$; $a_2 = 0.01269$; $b_2 = 61.55$; $c_2 = 13.59$; $a_3 = 0.006351$; $b_3 = 38.92$; $c_3 = 30.59$; $a_4 = 0.002171$; $b_4 = 73.55$; $c_4 = 55.27$.

Figure 5-13 shows the relative velocity histogram and its PDF estimate using KDE method and EM algorithm.

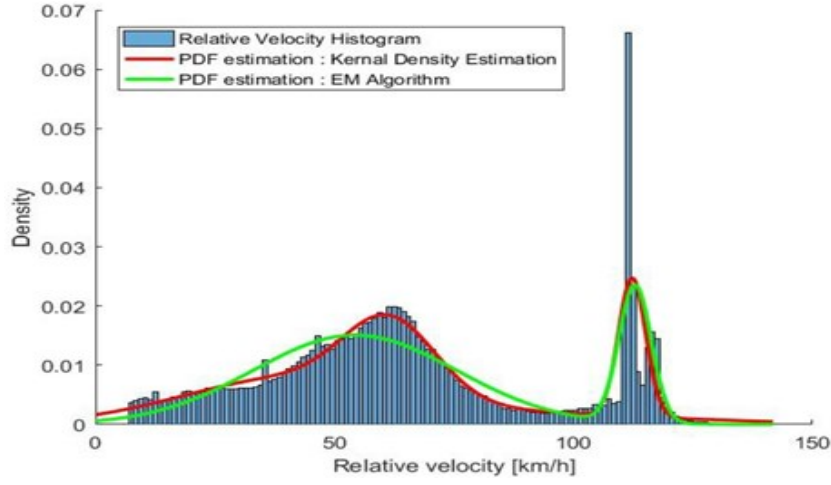


Figure 5-13 PDF estimation: Relative Velocity V_s

5.12.1.3 Temperature (weather)

$$P(x) = a_1 e^{-\left(\frac{(x-b_1)}{c_1}\right)^2} + a_2 e^{-\left(\frac{(x-b_2)}{c_2}\right)^2} \quad (5.15)$$

where $a_1 = 0.02373$; $b_1 = 18.45$; $c_1 = 6.127$; $a_2 = 0.02484$; $b_2 = 2.092$; $c_2 = 17.17$.

Figure 5-14 shows the temperature histogram and its PDF estimate using KDE method and EM algorithm.

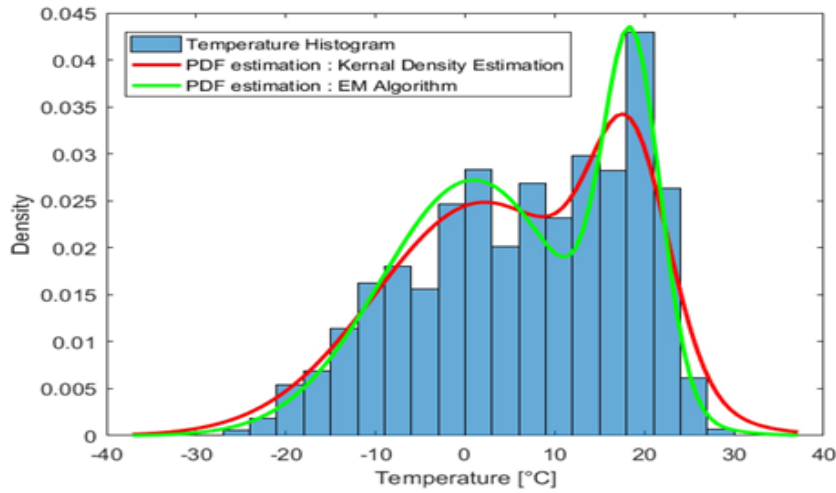


Figure 5-14 PDF estimation: Temperature

5.12.1.4 Humidity (weather)

$$P(x) = a_1 e^{-\left(\frac{(x-b_1)}{c_1}\right)^2} \quad (5.16)$$

where $a_1 = 0.02916$; $b_1 = 73.69$; $c_1 = 19.25$.

Figure 5-15 shows the humidity histogram and its PDF estimate using KDE method.

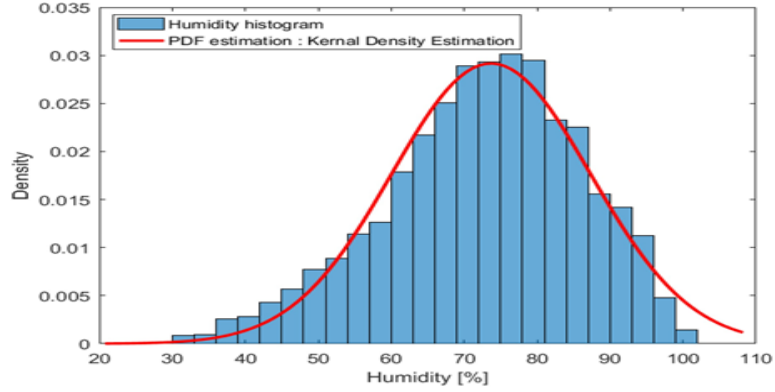


Figure 5-15 PDF estimation: Humidity

5.12.1.5 Visibility (weather)

$$P(x) = a_1 e^{-\left(\frac{(x-b_1)}{c_1}\right)^2} + a_2 e^{-\left(\frac{(x-b_2)}{c_2}\right)^2} \quad (5.17)$$

where $a_1 = 0.04065$; $b_1 = 22.49$; $c_1 = 9.321$; $a_2 = 0.03104$; $b_2 = 9.778$; $c_2 = 6.129$.

Figure 5-16 shows the visibility histogram and its PDF estimate using KDE method and EM algorithm.

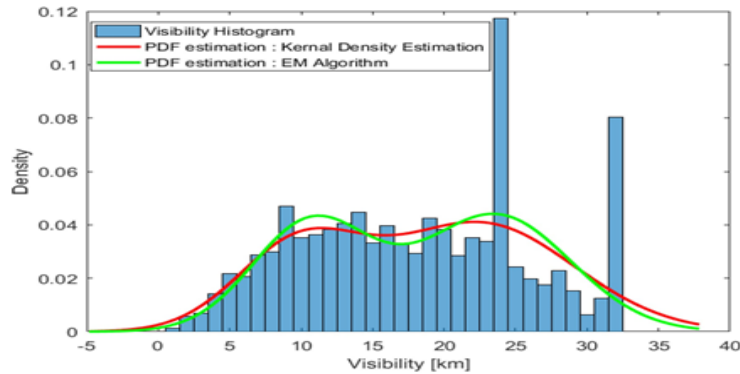


Figure 5-16 PDF estimation: Visibility

5.12.2 Metropolis-Hastings sampling results

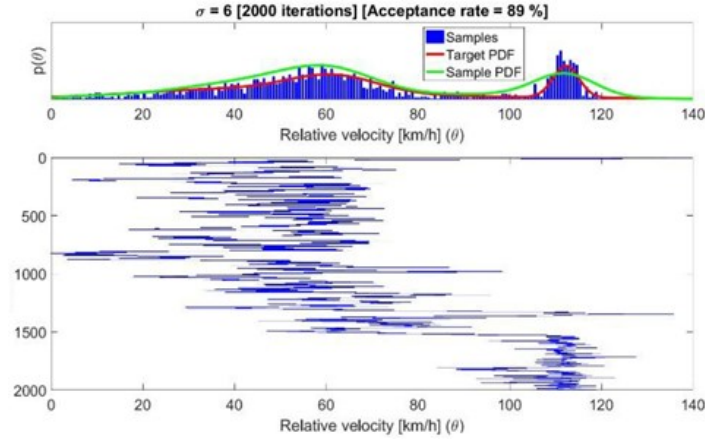


Figure 5-17 Sampling results ($\sigma = 6$): Relative Velocity (2000 iterations)

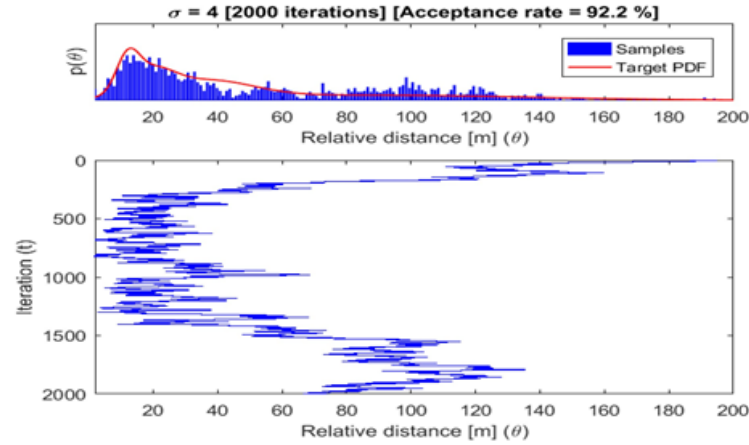


Figure 5-18 Sampling results ($\sigma = 4$): Relative Distance (2000 iterations)

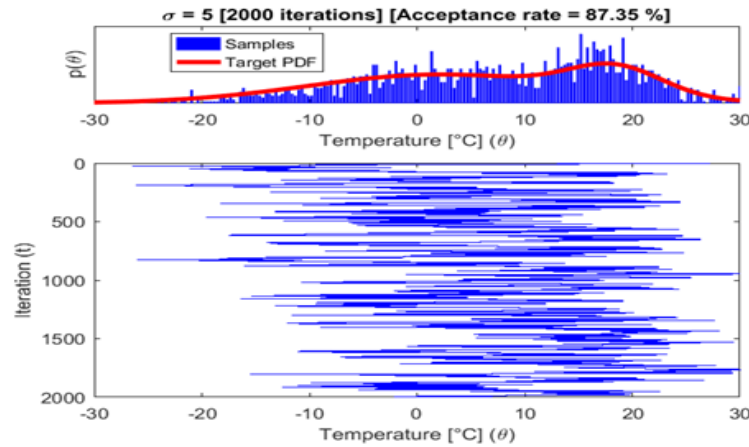


Figure 5-19 Sampling results ($\sigma = 5$): Temperature (2000 iterations)

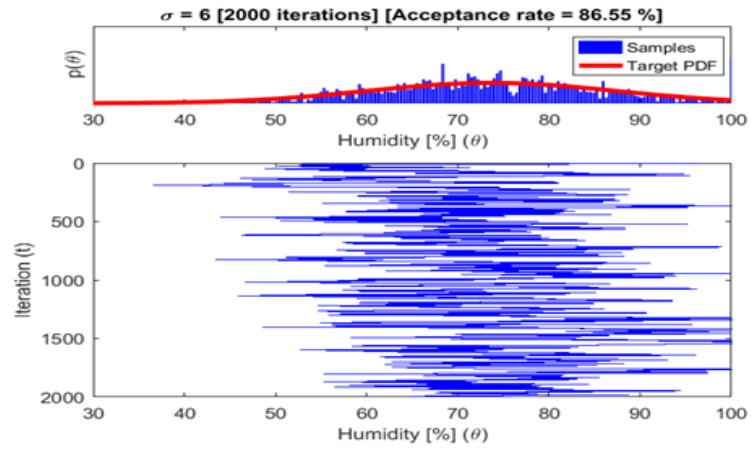


Figure 5-20 Sampling results ($\sigma = 6$): Humidity (2000 iterations)

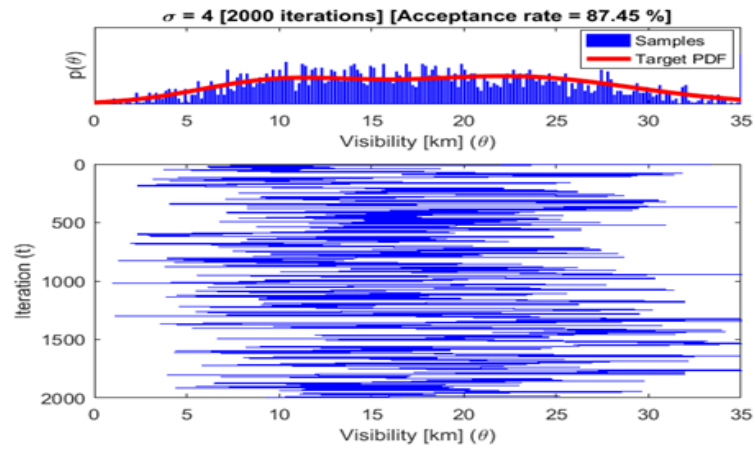


Figure 5-21 Sampling results ($\sigma = 4$): Visibility (2000 iterations)

Chapitre 6

Worst-case Scenarios Identification Approach for the Evaluation of Advanced Driver Assistance Systems in Intelligent/Autonomous Vehicles under Multiple Conditions

6.1 Avant-propos

6.1.1 Auteurs et affiliation

Nacer Eddine Chelbi : Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Membre du Laboratoire sur l'Intelligence Véhiculaire (LIV).

Denis Gingras : Professeur, Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Responsable du Laboratoire sur l'Intelligence Véhiculaire (LIV).

Claude Sauvageau : Directeur essais et immobilisations, PMG Technologies inc. Montréal, Canada.

6.1.2 Date de soumission

20 novembre 2018.

6.1.3 État de l'acceptation

Soumis.

6.1.4 **Revue**

Journal of Intelligent Transportation Systems: Technology, Planning, and Operations

6.1.5 **Titre français**

Approche d'identification des pires scénarios pour l'évaluation de systèmes avancés d'assistance à la conduite dans des véhicules intelligents/autonomes soumis à de multiples conditions

6.1.6 **Résumé français**

Pour démontrer les performances attendues d'un système avancé d'aide à la conduite (ADAS) dans un véhicule intelligent ou hautement automatisé, les approches de test doivent inclure des simulations, des tests sur piste et des tests sur route (tests sur le terrain / essais réels sur route). L'objectif principal de nos travaux est de proposer une nouvelle approche d'évaluation conduite par une entité externe dans laquelle le véhicule est traité comme une boîte noire. Cette approche permet à la fois d'identifier un ensemble de scénarios défavorables pour une application ADAS donnée et combine les trois approches de test mentionnées précédemment. La méthode d'évaluation proposée est divisée en trois parties: 1) synthèse des scénarios, stratégie d'échantillonnage et simulations, 2) évaluation et classification des risques, 3) validation. Dans ce chapitre, nous présentons notre approche proposée pour la partie validation. Nous l'avons décomposée en trois parties: une première partie, qui décrit les tests physiques du système de freinage automatique d'urgence (AEB) étudié et les différentes techniques d'apprentissage automatique et ensembliste utilisées pour créer son modèle prédictif. Une deuxième partie qui traite la base de données de tests opérationnels sur le terrain (SPMD), de la mise en œuvre de notre nouvelle stratégie d'échantillonnage basée sur l'algorithme Metropolis-Hastings original, modifié et multivarié. La troisième partie porte sur la collecte des résultats de prédiction, puis sur l'évaluation du risque de chaque test, pour les classer à l'aide d'une technique non supervisée (K-Means clustering), ce qui nous permet de construire un ensemble des pires scénarios et faire une sélection finale. Enfin, deux applications web (full stack) développées et déployées sont présentées.

6.2 Abstract

To demonstrate the expected performance of an advanced driver assistance system (ADAS) in an intelligent or a highly automated vehicle, test approaches should include a combination of simulations, track tests and road tests (Field Tests / Real Test Drives). The main objective of our work is to propose a new evaluation approach conducted by an external entity where the vehicle is treated as a black box. This approach allows both the identification of a set of worst-case scenarios for a given ADAS application, and combines the three test approaches mentioned earlier. Our proposed evaluation approach is broken down into three parts: 1) Scenarios synthesis, sampling strategy and simulations, 2) Risk assessment and classification, 3) Validation. In this article, we present our proposed approach for the validation part. We broke it down into three parts: A first part, which includes a description of the studied autonomous emergency braking (AEB) physical track tests and of the different machine and ensemble learning techniques that were used to create its predictive model. A second part that deals with the Field Operational Tests database (SPMD), the implementation of our new sampling strategy based on the original, modified and multivariate Metropolis-Hastings algorithm. The third part focuses on collecting the prediction results, then assess the risk of each test, to classify them using a non-supervised technique (K-Means clustering), and this allows us to build a set of worst-case scenarios to make a final selection. Finally, two developed and deployed full stack web applications are presented.

6.3 Introduction

Intelligent and autonomous vehicles (IAVs) are reaching unprecedented complexity, making their evaluation and validation even more complex. Advanced driver assistance systems (ADAS) are the basis for the development of IAVs [4] and they go through a validation process in several stages: development, demonstration and deployment [5]. The findings of several researchers suggest that in order to advance autonomous vehicles (AV) into daily use, alternative testing methods must be developed to supplement on-the-road testing. Alternative methods might include accelerated testing, virtual testing and simulators, mathematical modelling, scenario testing and pilot studies [5].

During the development phase, several approaches are used [6]: Model in the Loop (MIL), Software in the Loop (SIL), Processor in the Loop (PIL) and Hardware in the Loop (HIL). According to Pawel et al [7], model-driven development (MDD) and model-based testing (MBT) are the preferred approaches for the development of ADAS systems. An approach that usually needs an extensive understanding and knowledge about the system operation. The authors in [8] presented different X-in-the-loop (XiL) approaches and their validation against proving ground tests. They also give an apprehensive state of the art of different XiL and validation methods where several simulations and validation standards are listed. For a more efficient methodology in the vehicular domain and especially in the validation of ADAS and inter-vehicle communication systems, a new concept has been proposed: Vehicle Hardware in the Loop (VEHIL) by TNO Automobile [9]–[11]. The latter consists of integrating a vehicle into the validation loop, but it remains in a laboratory environment. There is consensus among experts that mastering the validation efforts in the future can only be achieved by means of significantly more simulations [8]. The authors in [12] stressed out, the importance of using a simulation strategy in the development process of active safety and advanced driver assistance systems where they illustrated both how to perform real time and non-real-time simulations on a multicore processor machine. ADAS validation shall address deterministic (safety concept) and non-deterministic aspects where they affirm that statistical testing is necessary by combining driving tests and simulations [13]. The evaluation and validation methods used are for the most deterministic. That means many factors, such as the speed of vehicles, are treated as if they were fixed and precisely known. They attempt to account for variability and uncertainty using fixed security factors. On the other hand, probabilistic approaches make it possible to quantify variations and uncertainties by using mainly probability distributions rather than fixed values. In this same approach, some initiatives have emerged in the field of intelligent and autonomous vehicles (IAVs) [14]–[17].

For the demonstration and deployment stages, two main approaches are used: closed-course tracks and on road testing (or Field Operation Tests). Closed-course tracks testing are usually performed in specialized test centres such as Transport Canada's Motor Vehicle Test and Research Centre, managed by PMG Technologies [18]. Field Operational Tests (FOT) are large scale testing programs designed to provide a more comprehensive assessment of the efficiency, quality, robustness and user acceptance of new vehicle technologies such as navigation,

communication systems, advanced driver assistance systems (ADAS) and cooperative systems, several projects have emerged over the last 20 years [19]–[23].

The development model V of ISO 26262, widely adopted by the automotive industry, establishes a framework that links each automotive test to a corresponding "specification" document. In [24], the authors discussed the challenges that must be overcome in order to be able to use this model to ensure that a given stand-alone vehicle is safe. Several procedures and approaches have been proposed for the development of ADASs, unfortunately, they can only be applied at a high level of abstraction in the development process; they do not provide objective requirements or evaluation criteria for the validation and calibration of ADAS and do not prescribe the use of specific tools and methods in the deployment process. In [5], the authors presented a framework defining how to measure safety of AVs (highly automated vehicles > level 4), in three stages: development, demonstration and deployment. In the deployment stage, they only include evaluation in public roads and exclude simulation, and closed course settings. Zhao et al. [177] categorize the existing AV evaluation approaches into four groups: test matrix, Naturalist-Field Operational Test (N-FOT), Monte Carlo simulations and Worst-Case Scenario Evaluation. To cope with the weaknesses of each approach, they proposed an Accelerated Evaluation approach based on Importance Sampling technique that can achieve an acceleration rate of hundreds to thousands of times greater compared to the original so-called Crude Monte Carlo method. However, this approach is only applicable before the release and deployment of AVs.

To demonstrate the expected performance of an advanced driver assistance system (ADAS) in an intelligent or a highly automated vehicle, test approaches should include a combination of simulations, track tests and road tests (Field Tests / Real Test Drives) [25]. IAVs will continue to make driving easier and especially safer. Nonetheless, the ultimate question remains what is the optimal evaluation approach to propose, which will have the ability to validate the behaviour and expected performance of embedded systems in intelligent and autonomous vehicles in all stages: development, demonstration and deployment. To our knowledge, most studies/proposals that tries to combine more than one approach are applied in the development stage.

The main objective of our work is to propose a new evaluation approach conducted by an external entity (in the deployment (or recall) stage), that do not have access to all information where the

vehicle is treated as a black box. This approach allows both the identification of a set of worst-case scenarios for a given ADAS application, and which combines the three test approaches mentioned earlier. Our proposed evaluation approach is broken down into three parts (see Figure 6-1): 1) Scenarios synthesis, sampling strategy and simulations, 2) Risk assessment and classification, 3) Validation. In this article, we present our proposed approach for the validation part. The first and second parts were discussed in [26]–[28]. However, all the needed information for understanding the validation part is well explained throughout this chapter.

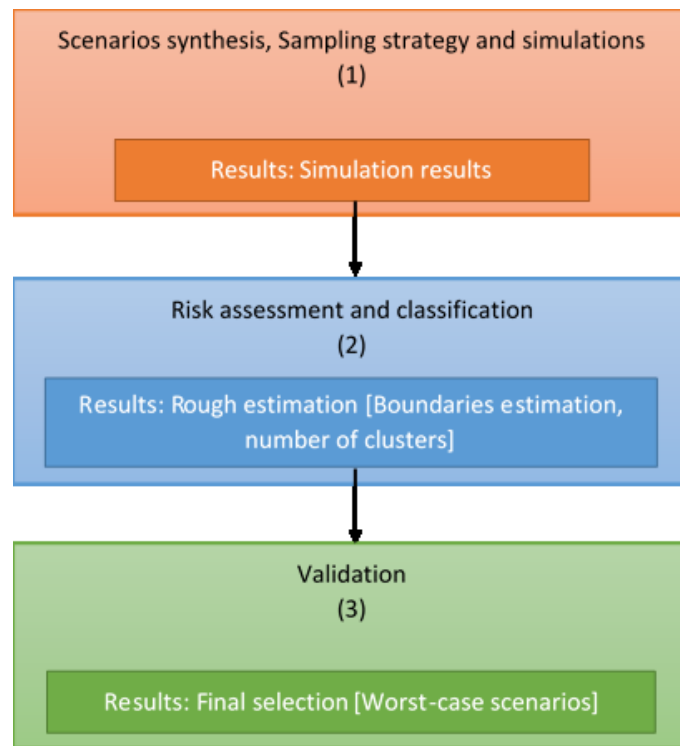


Figure 6-1 Proposed evaluation approach

6.4 Proposed validation approach

For a better visualization, here is the proposed validation approach architecture in Figure 6-2. We broke it down into three parts. A first part (in black), which includes a description of the studied autonomous emergency braking (AEB) track tests results and of the different machine learning techniques that were used to create its predictive model. A second part (in purple) that deals with the FOT database (SPMD), the implementation of a sampling strategy based on the original, modified and multivariate Metropolis-Hastings algorithm [26], [27]. The third part (in

red) focuses on collecting the prediction results, then assess the risk of each test, to classify them using a non-supervised technique (K-Means clustering), and this allows us to build a set of worst-case scenarios.

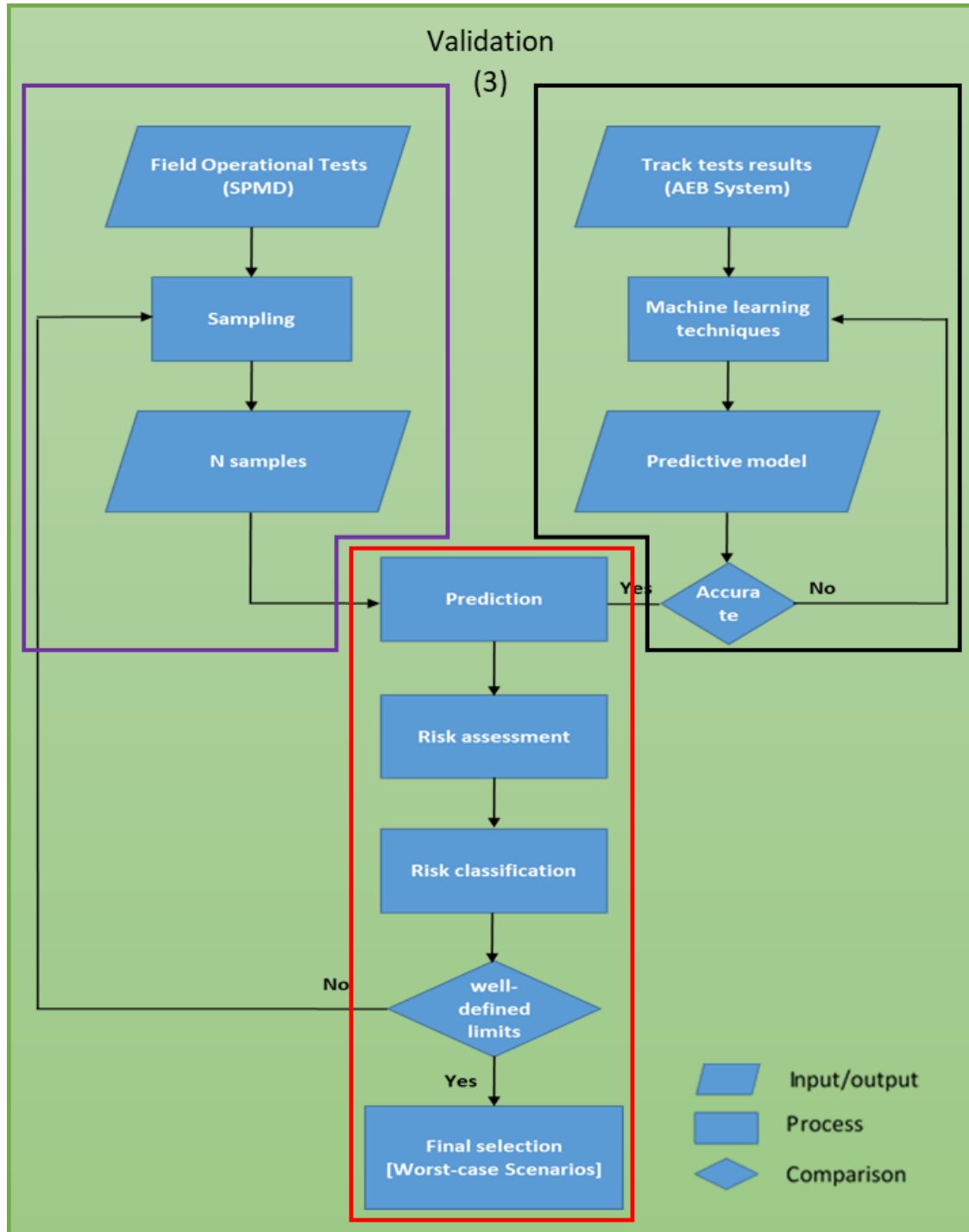


Figure 6-2 Proposed validation approach

6.5 Predictive model of the studied AEB system

6.5.1 Physical track tests

In this study, we used the track tests results of an AEB system conducted by PMG Technologies on behalf of Transport Canada. This test program aims to assess the performance and effectiveness of a forward crash avoidance and mitigation technology available on the market today. The term used for this technology is Autonomous Emergency Braking (AEB) which is composed of three (3) functionalities: Forward Collision Warning (FCW); Crash Imminent Braking (CIB) and; Dynamic Brake Support (DBS).

An AEB system is a proprioceptive predictive security system that aims to avoid or reduce the severity of rear-end collisions. The system uses information from sensors that monitor the area in front of the traveling car to identify objects with potential collision and apply an appropriate braking policy. The alerting and decision-making algorithms implemented in AEB systems often use measurements based on collision time (TTC: Time-to-Collision), the critical braking distance and the required deceleration. The system begins to warn the driver before the expected impact (TTC). At TTC, a fraction (between 40% and 60%) of the brake pressure is applied. This gives extra time for the driver to react and increases the possibility of braking or avoiding the collision. If the driver starts braking alone, the system will help him with extra strength, reaching the maximum braking value. If no reaction is taken, the full braking pressure is applied.

Much work has been carried out internationally in developing test procedures and targets to be used in AEB testing programs (vehicle and pedestrian). The work carried out in the test program [178] was based primarily on test procedures developed by the National Highway Traffic Safety Administration (NHTSA) [179], the International Organization for Standardization (ISO) [180] the European New Car Assessment Program (Euro NCAP) [181] as well as the Research Council for Automobile Repairs (RCAR). The targets to be used will include the Allgemeiner Deutscher Automobil Club (ADAC) – Euro NCAP Vehicle Target (EVT) and NHTSA – Strikeable Surrogate Vehicle (SSV). The specific objectives of this test program are as follows:

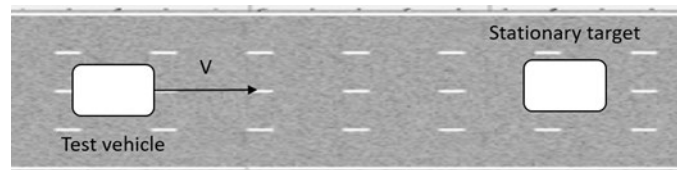
- Determine speed reduction and maximum speed at which collisions can be avoided for each vehicle;

- Identify the limitations in misses and false positives of each AEB systems; Assess test methods;
- Evaluate the suitability of the ADAC and SSV targets as vehicle targets for AEB systems; and
- Assess AEB functionality under different ambient conditions such as setting sun, darkness, fog, rain, ice and snow, if and when possible.

Each vehicle was tested with both the ADAC and SSV targets. Here is a summary of the different test scenarios.

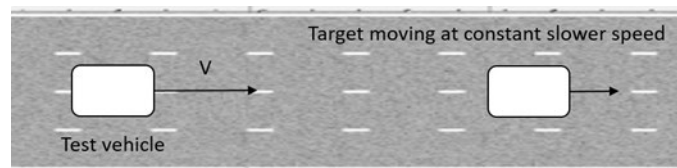
6.5.1.1 Stationary Target (ADAC and SSV)

This test determines whether the AEB system would react to a stationary target simulating, for example, a vehicle stopped in traffic or parked. The test vehicle approaches the stationary target at test speeds from 10 to 50 km/h. The speed is increased in 10 km/h increases if the system avoids a collision. If a collision occurs, the test is run at 5 km/h lower to establish the collision point in 5 km/h increments.



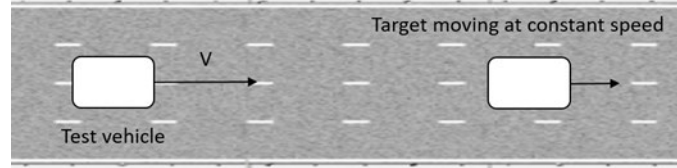
6.5.1.2 Target moving at a constant slower speed (ADAC and SSV)

This test evaluates how the AEB system reacts to a slower-moving vehicle. The test vehicle follows the target which is moving at a slower constant speed. It is performed at two (2) different speeds (see test matrix in Table 6-1) starting with 150 metre headway and with the speed stabilized, 100 metres from the target.



6.5.1.3 Target decelerating to a stop (ADAC and SSV)

This test evaluates how the AEB reacts to a vehicle ahead which is decelerating to a complete stop. The headway and deceleration rates are noted in the test matrix.



6.5.1.4 AEB track test matrix

Here are the speed, headway and deceleration rates for each scenario:

Table 6-1 Test matrix

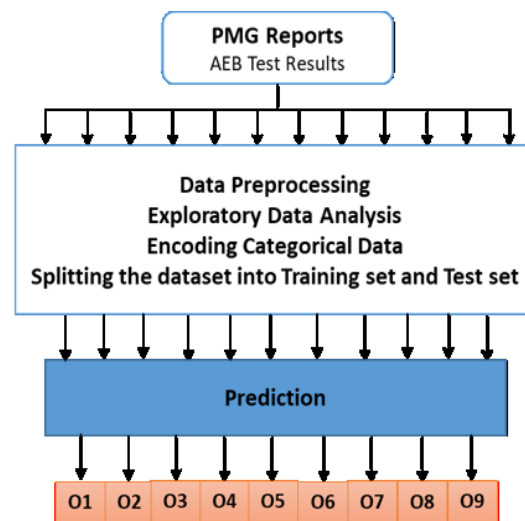
Pre-crash scenario	Test Vehicle Speed (km/h)	Target Speed (km/h)
A1. Stopped target	10, 20, 30, 40, 50 (-5 if impact occurs)	0
B1. Target moving at slower constant speed	40 (TTC=5s @ 34m)	16
B2. Target moving at slower constant speed	72 (TTC=5s @ 56m)	32
C1. Target decelerating to a complete stop	56 (14 m headway)	56 @ 0.3g

6.5.2 Predictive model construction

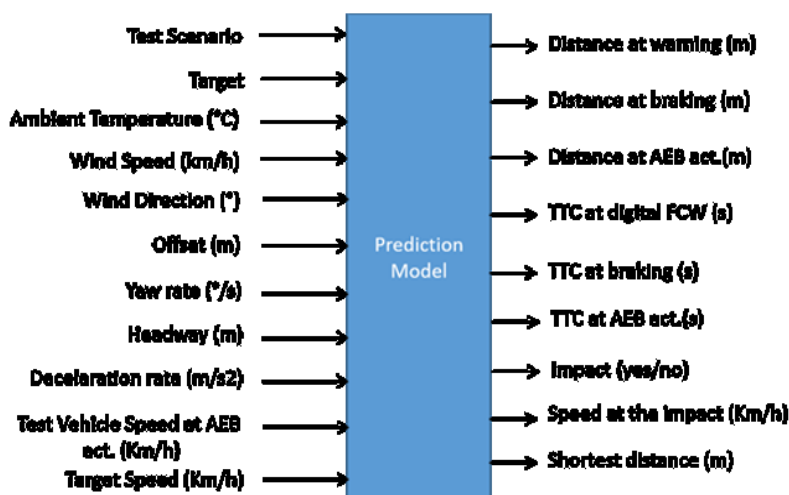
In this section, the goal is to build prediction models from the PMG physical test results. These models allow us both to interpolate the track test results, and in the proposed validation approach as will be discussed in this article. Figure 6-3-(a) shows a summary diagram of the procedure to be followed, and we will discuss briefly the used machine learning techniques in the next sections. For the identification, evaluation and validation of each model, the following steps have been followed:

- 1) Inputs/Outputs identification
- 2) Encoding categorical data

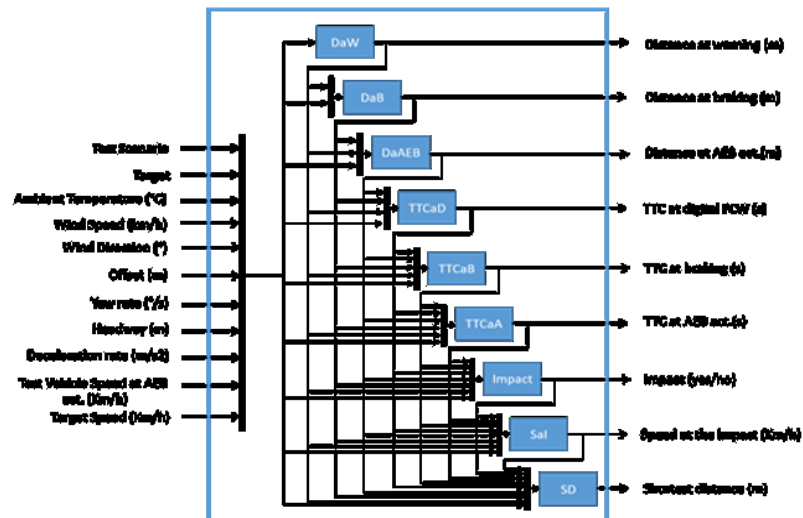
- 3) Splitting the dataset into training set (80%) and test set (20 %)
- 4) Building the model using the appropriate machine learning technique (see Table 6-3 below)
- 5) Predicting test set results
- 6) Evaluation
- 7) Validation (K-Fold Cross Validation)
- 8) Visualization (actual results vs predicted results)
- 9) Serializing and saving the model to disk (for further usage on the web applications (see section 7.11))



(a)



(b)



(c)

Figure 6-3 (a) Followed approach -(b) Inputs/outputs prediction model - (c) Summary diagram

6.5.2.1 Inputs/outputs of the AEB system

For this particular AEB system (the name of the vehicle is not disclosed for confidentiality reasons), 77 tests were conducted and 47 variables are saved at the end of each test. 74 is the number of valid tests. Of the 47 variables, only the most relevant ones to the studied AEB system are retained:

- Inputs: Test scenario [S], Target [Tar], Ambient temperature ($^{\circ}\text{C}$) [T], Wind speed (km/h) [W-S], Wind Direction ($^{\circ}$) [W-D], Offset (m) [Of], Yaw rate ($^{\circ}/\text{s}$) [Y-r], Headway (m) [Head], Deceleration rate (m/s^2) [Dec], Test Vehicle Speed at AEB activation (km/h) [T-V-S], Target Speed (km/h) [T-S].
- Outputs: Distance at warning (m), Distance at braking (m), Distance at AEB activation (m), TTC at digital FCW (s), TTC at braking (s), TTC at AEB activation (s), Shortest distance (m) (distance between target and test vehicle after full braking), Speed at the impact (km/h), Impact (yes/no).

In Figure 6-3-(b), we give an overview of the retained inputs and outputs of the AEB system, we see that we have eleven (11) inputs and nine (9) outputs. Our strategy consists of building a prediction model for each output. The predicted output of each model will be injected as input for future models. Here is a summary diagram in Figure 6-3-(c).

6.5.2.2 Encoding Categorical Data

From here we begin the data pre-processing phase. First of all, we take care of the missing data, then we encode the categorical data, to split the data set into training set and tests set and finally to apply feature scaling to put all of our variables on the same scale. We can see that in our data set, we have categorical variables (Test scenario, Target) and numerical variables. To encode these categorical data, we create what we call dummy variables. We take as an example the variable “Target”, we have two targets (ADAC and SSV). The idea here is to replace SSV and ADAC with a binary representation (see example in Table 6-2), ‘00’ for the first and ‘10’ for the second.

To distinguish the effects of each dummy variable and not have a variable duplication, it is important to eliminate one variable. Otherwise, the regression techniques will not work properly. This is called avoiding the dummy variable trap [182].

Table 6-2 Example: encoding categorical data

Target	Categorical data	After encoding
ADAC	ADAC	10
SSV	SSV	00

6.5.2.3 Splitting the dataset into training set (70 to 80%) and test set (20 to 30%)

In any machine learning technique, the dataset is usually split into training set and tests set. The training set is the data that the model will learn from in order to be generalized to other data later on. The test set on which we test the performance of this machine learning model and the performance on the test set should not be that different from the performance on the training sets.

6.5.2.4 Building the Models

For each output and to improve the accuracy of its prediction model, an appropriate machine learning technique was used. We used a wide range of techniques and here is the exhaustive list in Table 6-3.

At the beginning we did not get a satisfactory result for the Impact output prediction model. And to improve its accuracy, several machine learning techniques were investigated and then used to produce a majority voting based classifiers.

So the idea consists of combining several classifiers via majority vote (see Figure 6-4) in which we predict the final class label as the class label that has been predicted most frequently by the other classification models [64]. This way, we will combine strengths of all used classification techniques, in this study: Logistic Regression, K-Nearest Neighbours (K-NN), Suport Vector Machine (SVM), Kernel Suport Vector Machine (K-SVM), Naive Bayes Calssifier (NB), Decision Trees Classifier, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Ensemble Learning : Bagging [Boot- strap Aggregating], Ensemble Learning : Bagging [Random Forest], Ensemble Learning : Boosting [AdaBoost (Adaptive Boosting)], Ensemble Learning : Stacking [Standard Stacking], Ensemble Learning : Stacking [Stacking Cross-Validation].

Table 6-3 Used machine learning techniques

Output	Used machine learning (ML) technique	Type
Distance at warning (m)	Multiple Linear Regression	Regression
Distance at braking (m)	Multiple Linear Regression	Regression
Distance at AEB act (m)	Multiple Linear Regression	Regression
TTC at digital FCW (s)	Multiple Linear Regression	Regression
TTC at braking (s)	Multiple Linear Regression	Regression
TTC at AEB act (s)	Multiple Linear Regression	Regression
Impact (yes/no)	- Logistic Regression	Classification
	- Modified multiple linear regression	
	- K-Nearest Neighbours (K-NN)	
	- Support Vector Machine (SVM)	
	- Kernel Support Vector Machine (K-SVM)	
	- Naive Bayes Classifier (NB)	
	- Decision Trees Classifier	
	- Linear Discriminant Analysis (LDA)	
	- Quadratic Discriminant Analysis (QDA)	
	- Ensemble Learning : Bagging [Bootstrap Aggregating]	
	- Ensemble Learning : Bagging [Random Forest]	
	- Ensemble Learning : Boosting [AdaBoost (Adaptive Boosting)]	
	- Ensemble Learning : Stacking [Standard Stacking]	
	- Ensemble Learning : Stacking [Stacking Cross-Validation]	
	- Ensemble Learning : Majority Voting	
Speed at impact (km/h)	Support Vector Regression	Regression
Shortest distance (m)	Gaussian Process Regression	Regression

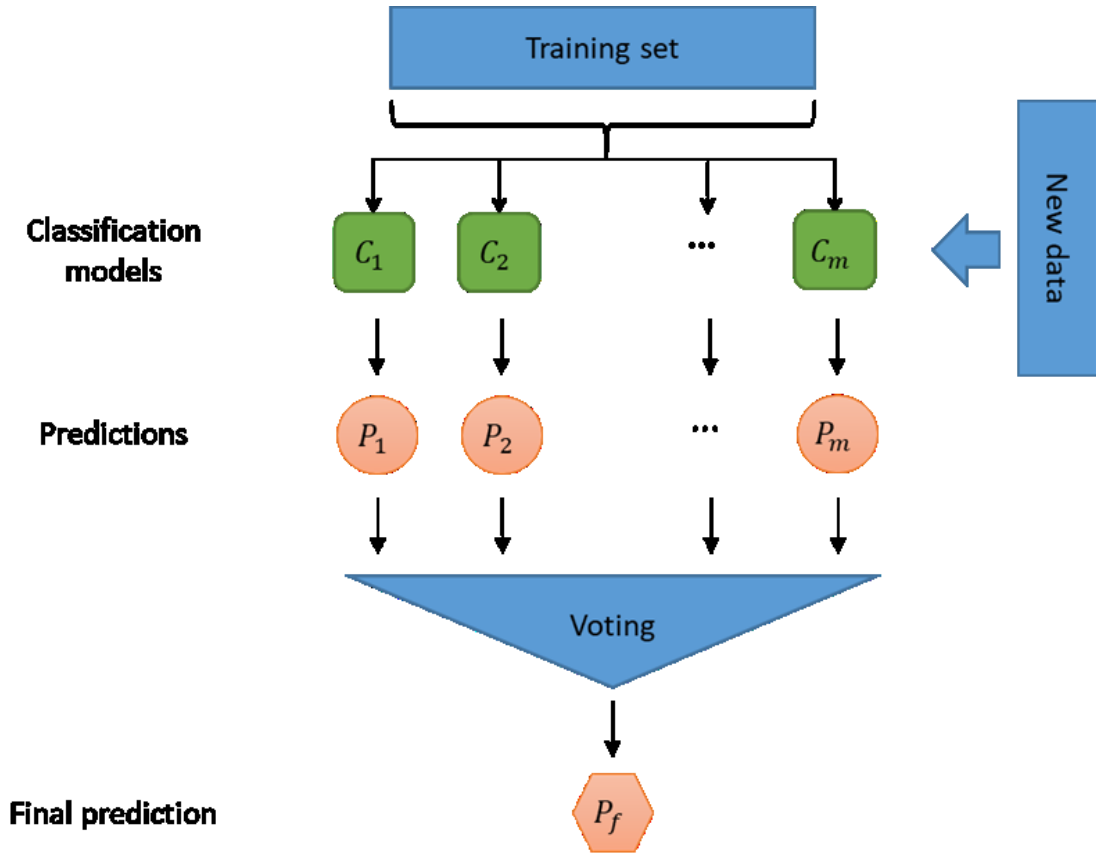


Figure 6-4 Ensemble Learning: Majority Voting Classifier [64]

The basic idea is to predict the class label \hat{y} via majority voting of each classifier C_j :

$$\hat{y} = \text{mode}\{C_1(x), C_2(x), \dots, C_m(x)\} \quad (5.18)$$

As an example, we assume that we are using five classifiers that classify a training sample as follows: classifier 1 \rightarrow class 0, classifier 2 \rightarrow class 0, classifier 3 \rightarrow class 0, classifier 4 \rightarrow class 1, classifier 5 \rightarrow class 1:

$$\hat{y} = \text{mode}\{0, 0, 0, 1, 1\} \quad (5.19)$$

Via majority vote, we would classify the sample as «class 0».

6.5.2.5 Predicting test set results

After the training step and the creation of the model is done, in this step, we see how this model will predict some new observations (test set).

6.5.2.6 Evaluation and Validation (K-Fold Cross Validation)

For regression models we used multiple metrics such as: R², RMSE, mean of the K- Fold cross validation scores, explained variance; and RMSE, accuracy, precision, recall, f1-score, and ROC curves for the classification models.

- R² (coefficient of determination): It provides a measure of how well future samples are likely to be predicted by the model and is calculated as follows [183]:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{samples}-1} (y_i - \bar{y})^2} \quad (5.20)$$

- RMSE (Root Mean Squared Error): measures how much error there is between two data sets and it is calculated as follows [183]:

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2} \quad (5.21)$$

- Mean K-Fold cross validation score: *«In k-fold cross-validation, the original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once»* [184].
- Explained variance score: it provides a measure of how well the model accounts for the variation of a given dataset and is calculated as follows [183]:

$$explained - variance(y, \hat{y}) = 1 - \frac{Var\{y - \hat{y}\}}{Var\{y\}} \quad (5.22)$$

- Accuracy: it provides the proportion of correct classifications (true positives (TR) and negatives (TN)) from the overall number of cases and is calculated as follows [185]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.23)$$

- Precision: it provides the proportion of correct positive classifications (true positives) from cases that are predicted as positive (true positives and false positives (FP)) and is calculated as follows [185]:

$$Precision = \frac{TP}{TP + FP} \quad (5.24)$$

- Recall: it provides the proportion of correct positive classifications (true positives) from cases that are actually positive (true positives and false negatives (FN)) and is calculated as follows [185]:

$$Recall = \frac{TP}{TP + FN} \quad (5.25)$$

- F1-score: it provides a weighted average of the precision and the recall and is calculated as follows [185]:

$$F1_score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (5.26)$$

- ROC curve (Receiver operation characteristics): is a technique for visualizing, organizing and selecting classifiers based on their performance (ROC). The quality of a ROC curve is often summed up by a single number using the area under the curve (AUC). The higher the AUC is better; the maximum is 1 [185].

6.5.2.7 Serializing the machine learning models

In this part, we save our models to specific binary files (using the python joblib library). This will allow us to load it later in order to make new predictions or to be used in a deployed web application (see section 6.11).

6.5.2.8 Results and Discussions

The prediction results of the first 6 outputs: distance at warning, distance at braking, distance at AEB act, TTC at digital FCW, TTC at braking and TTC at AEB act, are satisfactory with

accuracies that exceed 96% (between 96 and 99% - K-Fold cross validation). Here is a summary in Table 6-4 of the regressions results:

Table 6-4 Prediction results of the first 6 outputs

Output	RMSE	R²	Mean K-Fold cross validation score	Explained variance score
Distance at warning (m)	0.77	0.95	0.96	0.97
Distance at braking (m)	0.36	0.98	0.98	0.99
Distance at AEB act (m)	0.20	0.99	0.99	0.99
TTC at digital FCW (s)	0.13	0.98	0.85	0.86
TTC at braking (s)	0.01	0.99	0.99	0.99
TTC at AEB act (s)	0.01	0.99	0.99	0.99

For the Impact (yes/no) output, here is a summary in Table 6-5 of the classification results (all the results are the means of the K-Fold Cross Validation).

We used : RMSE, accuracy, precision, recall, f1-score, and ROC curves to assess the prediction models. Each technique has its own advantages and disadvantages. We have chosen AUC as our defining metric because it is often preferred over accuracy for binary classification. The main idea was to find and select the best classifier based on their performances (ROC). The rule is the higher the AUC is better.

We notice that the majority voting classifier performance outperform every other classifier with an AUC = 0.91 and an RMSE = 0.24, thus its selection as the main classifier (predictor) for the Impact (yes/no) output. In Figure 6-5-(a), we plot the performance characteristic (ROC curve) of each classifier.

Table 6-5 Classification results of Impact (yes/no) output

Used ML technique	RMSE	Accuracy	Precision	Recall	F1-score	AUC
Logistic Regression	0.40	0.81	0.85	0.81	0.81	0.87
Modified multiple linear regression	0.35	0.82	0.84	0.82	0.82	-
K-Nearest Neighbours (K-NN)	0.36	0.83	0.87	0.83	0.83	0.77
Support Vector Machine (SVM)	0.31	0.85	0.84	0.85	0.84	0.85
Kernel Support Vector Machine (K-SVM)	0.28	0.83	0.83	0.86	0.83	0.14
Naive Bayes Classifier (NB)	0.55	0.65	0.90	0.65	0.71	0.83
Decision Trees Classifier	0.40	0.81	0.83	0.81	0.80	0.66
Linear Discriminant Analysis (LDA)	0.35	0.82	0.81	0.82	0.81	0.80
Quadratic Discriminant Analysis (QDA)	0.41	0.83	0.69	0.83	0.75	0.50
Ensemble Learning : Bagging [Bootstrap Aggregating]	0.28	0.88	0.85	0.88	0.85	0.88
Ensemble Learning : Bagging [Random Forest]	0.31	0.86	0.75	0.86	0.80	0.80
Ensemble Learning : Boosting [AdaBoost (Adaptive Boosting)]	0.46	0.78	0.82	0.78	0.78	0.79
Ensemble Learning : Stacking [Standard Stacking]	0.26	0.88	0.82	0.88	0.84	0.69
Ensemble Learning : Stacking [Stacking Cross-Validation]	0.34	0.86	0.80	0.86	0.82	0.12
Ensemble Learning : Majority Voting	0.24	0.87	0.80	0.87	0.83	0.91

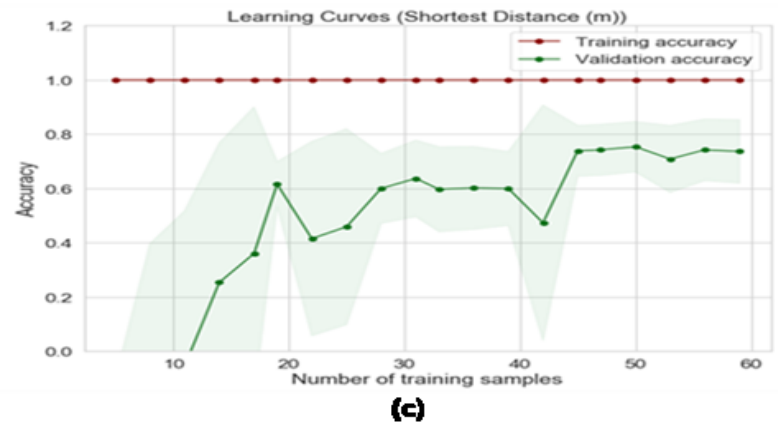
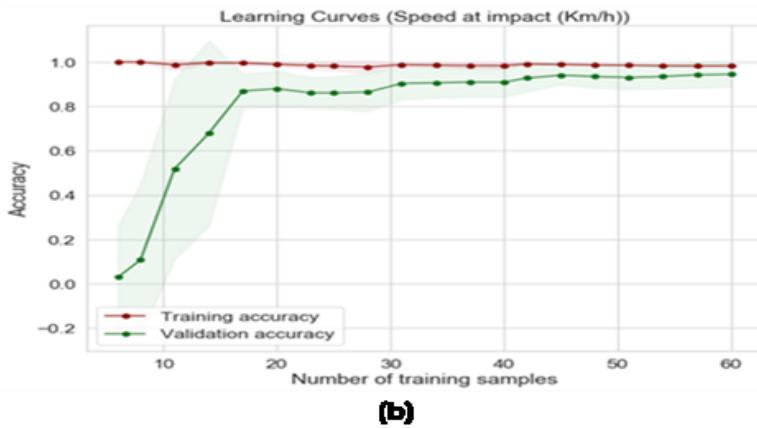
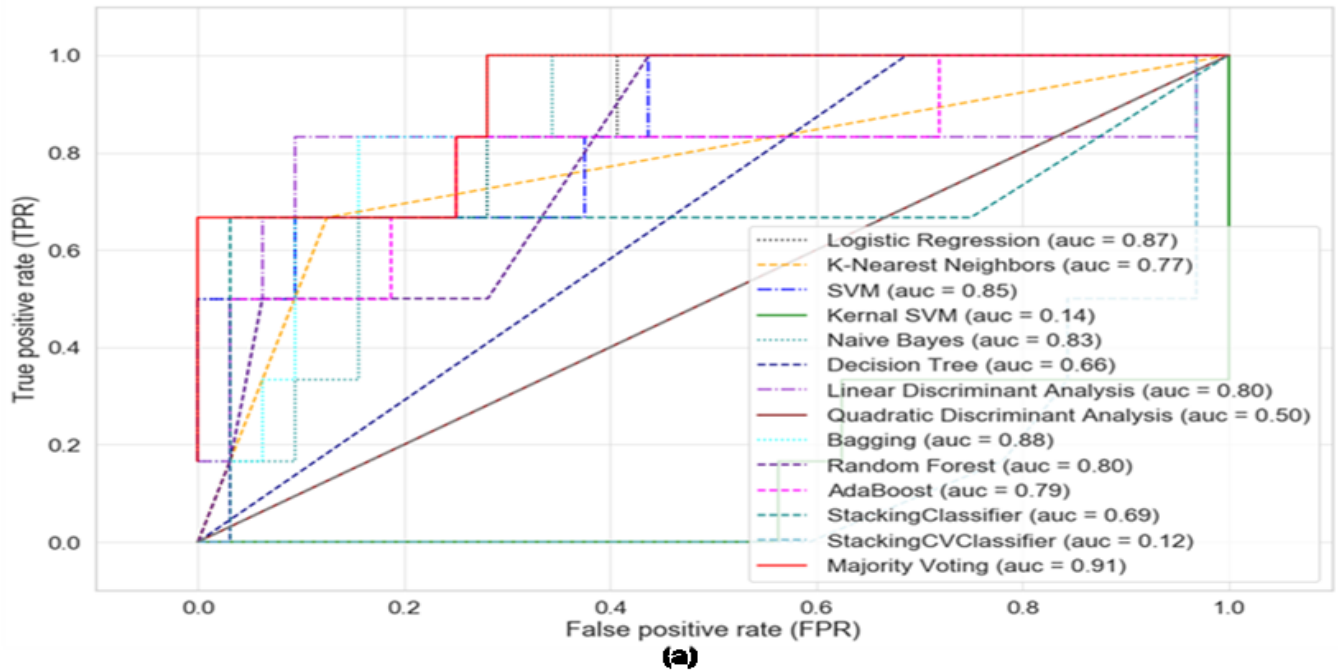


Figure 6-5 (a) ROC Curve of each classifier - Learning Curves: (b) Speed at impact - (c) Shortest Distance

For the prediction results of the last 2 outputs: speed at impact, shortest distance, here is a summary in Table 6 of the prediction results:

Table 6-6 Prediction results of the last 2 outputs

Output	RMSE	R ²	Mean K-Fold cross validation score	Explained variance score
Speed at impact (km/h)	1.22	0.95	0.95	0.95
Shortest distance (m)	0.63	0.63	0.73	0.67

We notice that the prediction results of the speed at impact output are satisfactory with an accuracy = 95%, which is not the case for the shortest distance output with an accuracy = 73%. Since this output is predicted in case of non-collision, its prediction accuracy is not as important, i.e. when we don't have a collision it means that we are 100% safe and whether the vehicle stopped at 1 m or 0.73 m is not that important. Furthermore, in risk classification (section 7), we are interested exclusively in test cases with collisions. For further investigations, we use learning curves to diagnose whether our learning algorithm need more training data or not.

In order to diagnose overfitting (high variance) or underfitting (high bias) of our models, the concept of Learning Curves is introduced [186]. By plotting the model training and validation accuracies as functions of the training set size, we can easily detect whether the model suffers from high variance or high bias, and whether the collection of more data could help address this problem. In our case, it allows us to decide whether an additional number of physical tests is necessary or not.

In the learning curves in Figure 6-5-(b) (Speed at impact), we can see that our model performs quite well ($> 95\%$ accuracy) on both the training and validation dataset if it had seen more than 45 samples during its training. We can easily confirm that the used number of physical tests is sufficient to predict this output (speed at impact) with an accuracy equal or higher than 95% (60 represents the training set size, 80% of 74 tests).

Which is not the case for the output "shortest distance", the learning curves in Figure 6-5-(c) shows that our model suffers from high variance (overfitting), which is indicated by the large gap between the training and validation accuracy. To solve this problem, we need to collect more training data (more physical tests) or to reduce the complexity of the model.

6.6 Sampling the FOT Database

The same sampling strategy presented in our article "New Field Operational Tests Strategy based on Metropolis-Hastings Algorithm [27]" has been applied in this section. For the Field Operational Tests (FOT) database, we used the Safety Pilot Model Deployment (SPMD) program. A research initiative highlighting the application of connected vehicle safety technologies and 'V2V, V2I (DSRC)' systems in real driving conditions. This program recorded 2842 equipped vehicles in Ann Arbor, Michigan for more than 2 years. The data needed to be

extracted are the inputs of the AEB system: Test Scenario, Target, Ambient Temperature, Wind Speed, Wind Direction, Offset, Yaw rate, Headway, Deceleration rate, Vehicle Speed Test at AEB act and Target Speed (see Figure 3-(a)(b)).

For discrete event parameters (Test Scenario and Target), their discrete distributions are obtained from the PMG database. For the sampling part, it is our modified version of the Metropolis-Hastings algorithm that has been applied (for more details check out "Modified Metropolis-Hastings Algorithm for Efficient Sampling from Discrete Probability Distributions (MMH-DPD) [26]").

For meteorological data, we opted for Meteoblue website [187]. A website that gives access to weather conditions from all over the world. The used data in this study are the meteorological data of Montreal (Canada) from 1 August 2008 to 31 July 2017.

6.6.1 Multivariate Metropolis-Hastings algorithm (Component-Wise Updating)

In previous applications ([26]–[28]) we used of the Metropolis-Hastings algorithm, and we focused on how to sample from univariate target distributions. However, in this study (validation part), we discuss its application to multivariate distributions (several dimensions). There are two well-known Metropolis-Hastings update procedures to sample from distributions in multiple dimensions: block-wise and component-wise. In the block-wise update procedure, the proposal distribution has the same dimensionality as the target distribution [188]. Its main problem is when the number of dimensions becomes large, it becomes more difficult to find a suitable proposal distribution.

Therefore, we were interested in the Component-Wise Updating procedure and we applied it to a bivariate distribution of Test Vehicle Speed and Headway parameters. In this approach, not like block-wise update procedure, the proposal distribution is univariate and the algorithm will loop over the dimensions in sequence [188]. Figure 6-6 shows on the right the real distribution and on the left the result of the sampling (250000 samples).

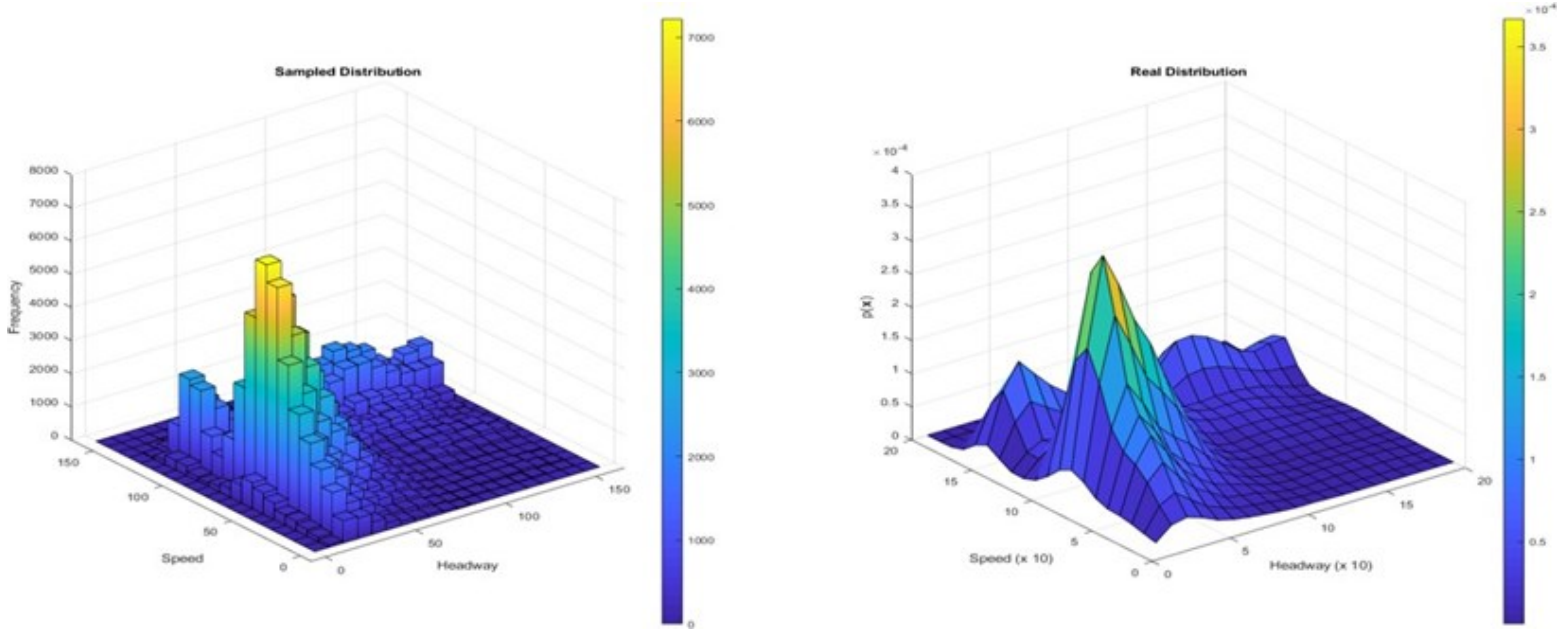


Figure 6-6 Sampled vs Actual bivariate distribution [Speed and Headway]

Here is a summary of the Component-Wise Updating Metropolis- Hastings Algorithm:

Component-Wise Updating Metropolis-Hastings Algorithm [188]:

- 1) Set $t = 1$
 - 2) Generate an initial state X^0
 - 3) Repeat until $t = N$
 - Set $t = t + 1$
 - For each dimension $i = 1..D$
 - Generate a proposal state x_i^* from $q(x_i|x_i^{(t-1)})$
 - Calculate the proposal correction factor: $c = \frac{q(x_i^{(t-1)}|x_i^*)}{q(x_i^*|x_i^{(t-1)})}$
 - Calculate the acceptance probability: $\alpha = \min\left(1, \frac{p(\theta^*)}{p(\theta^{(t-1)})} \times c\right)$
 - Draw μ from a uniform distribution: $Uniform(0,1)$
 - If $\mu \leq \alpha$, accept the proposal state x_i^* and set $x_i^{(t)} = x_i^*$
 - Else set $x_i^{(t)} = x_i^{(t-1)}$
-

6.6.2 Sampling results

The sampling results from the Test Scenario and the Target speed distributions are shown in Figure 6-7-(a) and (b) respectively. The remaining results can be found in Appendix. (Section 6.13.1). Each figure shows the sampling result where 10000 samples were drawn from the distribution of each parameter using the Metropolis-Hasting sampler (modified [26] for discrete events and the original for continuous events [27]). We have two plots in each figure, the top one shows the true target distribution (in red), along with a histogram of the samples obtained using the Metropolis-Hastings sampler. The lower panel plots the actual Markov chain: the sequence of generated values.

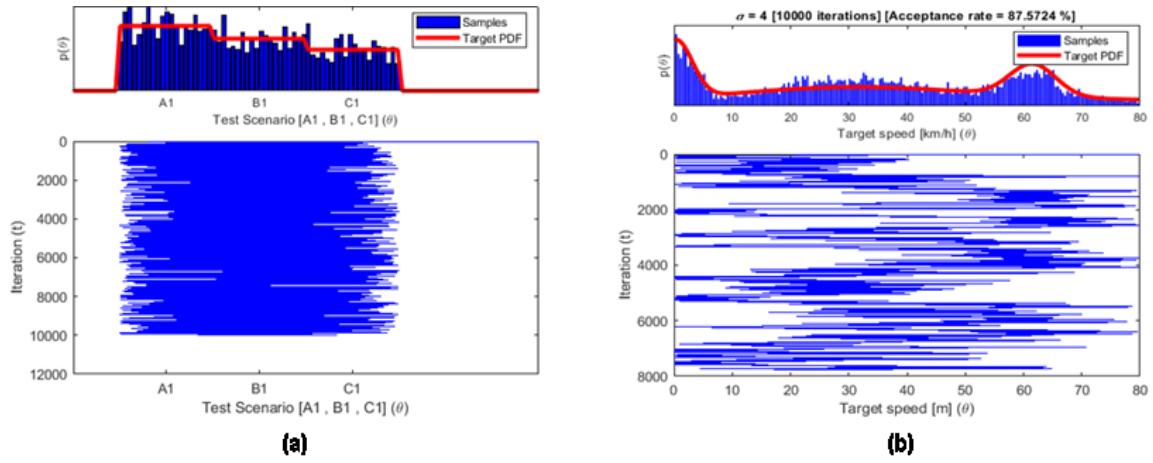


Figure 6-7 Sampling results: (a) Test Scenario (10000 samples) - (b) Target speed (10000 samples)

6.7 Prediction step

Then and as shown in Figure 6-2, the next step is to take N samples of each input parameter [Test Scenario, Target, Ambient Temperature, Wind Speed, Wind Direction, Offset, Yaw Rate, Headway, Deceleration rate, Test Vehicle Speed at AEB act, Target Speed], to provide them to the prediction model, then collect the predictions results (of each output) [Distance at warning, Distance at braking, Distance at AEB act, TTC at digital FCW, TTC at braking, TTC at AEB act, Impact (yes / no), Speed at Impact, Shortest distance], and finally, calculate the impact speed reduction rate. Here is a summary of the prediction part:

- 1) Choose the number of samples N and put $s = I$
- 2) Repeat until $s = N$
 - Take a sample of each input parameter
 - Provide the samples to the prediction model
 - Collect the prediction results
 - Calculate the impact speed reduction rate of each test
 - $s = s + I$

Table 6-7 below shows some samples provided to the prediction model. Unrealistic scenarios will be rejected in the final selection (example: test 8 will be rejected since target speed and deceleration in an A1 scenario need to be equal to zero).

Table 6-7 Examples of samples provided to the prediction model

N	S	Tar	T (°C)	W-S (km/h)	W-D (°)	Of (m)	Y-r (°/s)	Head (m)	Dec (m/s ²)	T-V-S (km/h)	T-S (km/h)
1	B1	SSV	15.58	45.78	333.66	1.39	2.09	38.90	-0.01	32.76	26.00
8	A1	ADAC	-3.63	41.29	309.06	0.77	1.47	28.37	-0.56	24.77	27.61
19	C1	SSV	-16.39	33.45	267.35	0.01	0.71	17.89	-1.8	31.92	12.30

We started by taking 10000 samples. We will increase the number of samples depending on whether the boundaries between risk levels are distinct or not (see Figure 6-2, Figure 10). Table 8 summarizes some descriptive statistics of the sampled data set.

Table 6-8 Descriptive statistics of the data set [10000 samples]

	T (°C)	W-S (km/h)	W-D (°)	Of (m)	Y-r (°/s)	Head (m)	Dec (m/s ²)	T-V-S (km/h)	T-S (km/h)
Count	10000	10000	10000	10000	10000	10000	10000	10000	10000
Mean	6.58	10.70	210.34	0.01	0.11	54.44	-0.53	72.35	35.01
Std	12.94	7.27	91.49	0.35	0.49	35.50	0.55	29.60	22.62
Min	-37.64	0.01	0.01	-1.14	-1.82	15.00	-4.11	0.00	0.00
50%	7.75	8.70	231.61	0.01	0.09	40.96	-0.35	68.72	34.66
75%	17.02	14.90	280.67	0.24	0.29	79.99	-0.16	99.82	56.56
Max	44.19	46.17	389.48	1.39	3.35	162.80	-0.03	138.15	79.97

6.7.1 Prediction results

For each test, the impact speed reduction rate is calculated as follows:

$$T_r = \frac{V_{vehicle} - V_{impact}}{V_{vehicle}} \times 100\% \quad (5.27)$$

The table below shows some examples of data collected and calculated after each prediction:

Table 6-9 Examples of data collected and calculated after each prediction

	Distance at warning (m)	Distance at braking (m)	Distance at AEB act. (m)	TTC at digital FCW (s)	TTC at braking (s)	TTC at AEB act. (m)	Impact (yes/ no)	Speed at the impact (km/h)	Shortest distance (m)	Reduction rate (%)
24	29.75	21.50	19.19	4.66	1.62	1.59	No Impact	0.00	0.0	100.00
25	34.87	25.04	21.91	5.68	1.80	1.68	Impact	2.00	0.0	93.22
26	31.26	22.42	19.67	5.39	1.72	1.57	Impact	2.45	0.0	90.50
27	28.37	21.13	18.96	4.10	1.57	1.59	No Impact	0.00	0.0	100.00

6.8 Risk assessment

To assess the risk of each test, the same scoring system mentioned in [28] is applied. In summary, for each test and in case of an impact, a proportional score to the reduction rate is attributed. The calculation of this score is given by the following formula:

$$Note^{(s)} = \left(\frac{V_{vehicle(s)} - V_{impact(s)}}{V_{vehicle(s)}} \right) \times points_{test(s)} \quad (5.28)$$

With $points_{test}$: Number of points available for different speeds. And it is deduced from the scoring table mentioned in Figure 6-8. With: Green = 1.000; Yellow = 0.750; Orange = 0.500; Brown = 0.250; Red = 0.000.

The scoring table was inspired by the Euroncap AEB systems TEST PROTOCOL [189]. We extrapolated the colour scheme from the scoring table of the AEB City to higher test speeds (>80 km/h). The scoring standard is a function of three factors: vehicle speed, impact speed and

points of the test. The calculation of the later (points of the test) is based on the frequency of accidents, where the most frequent accidents are more weighted than others.

As an example, we apply our calculation on the data in line 25 of Table 6-9. With: Vehicle test speed: $V_{vehicle(25)} = 29.48 \text{ km/h}$, impact speed: $V_{impact(25)} = 2 \text{ km/h}$, $points_{test(25)}=1$. So the attributed score for test number 25 is equal to 0.93.

$$Note^{(25)} = \left(\frac{V_{vehicle(25)} - V_{impact(25)}}{V_{vehicle(25)}} \right) \times points_{test(25)} = \left(\frac{29.48 - 2}{29.48} \right) \times 1 = 0.93 \quad (5.29)$$

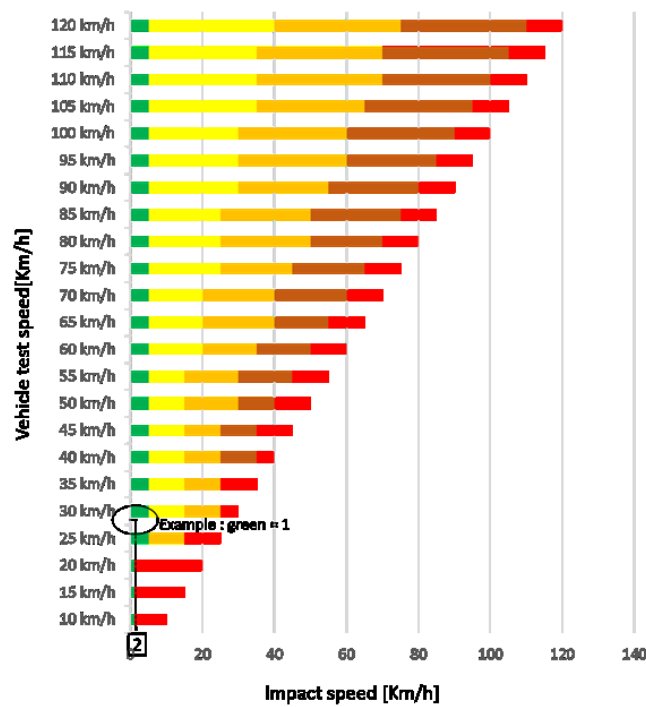


Figure 6-8 Scoring table

6.8.1 Risk assessment results

We repeat the same approach with the remaining tests. Table 6-10 shows the scoring results of some tests.

Table 6-10 Examples of some scoring results

Distance at	Distance at	Distance at AEB act.(m)	TTC at digital	TTC at braking (s)	TTC at AEB act.(m)	Impact (yes/no)	Speed at the	Shortest distance (m)	Reduction rate (%)	Notes
-------------	-------------	-------------------------	----------------	--------------------	--------------------	-----------------	--------------	-----------------------	--------------------	-------

	warning (m)	braking (m)		FCW (s)				impact (km/h)			
24	29.75	21.50	19.19	4.66	1.62	1.59	No Impact	0.00	0.0	100.00	1
25	34.87	25.04	21.91	5.68	1.80	1.68	Impact	2.00	0.0	93.22	0.93
26	31.26	22.42	19.67	5.39	1.72	1.57	Impact	2.45	0.0	90.50	0.90
27	28.37	21.13	18.96	4.10	1.57	1.59	No Impact	0.00	0.0	100.00	1

6.9 Risk classification

In a classification problem where labels are not known beforehand, an unsupervised learning technique (clustering) is often used. Similar data points, according to some notions of similarity, are considered in the same group. We call these groups clusters.

In our study, we applied the K-Means Clustering algorithm; a simple classification algorithm that is widely used in the literature whether in the academic field or in industry. One of the disadvantages of this algorithm is that it is necessary to specify the number of clusters " k " a priori. Several methods are often used to choose the correct number of clusters " k ": the elbow method, silhouette method, and the G-means algorithm, etc. Given the value of k , the algorithm tries to find the k clusters in the dataset. In our case, we use the results of the second part, explained in [28] and mentioned in Figure 6-1, as a starting point for the number of initial clusters (at least 4 clusters).

6.9.1 Case Study 1: 10000 Samples

The starting data set consists of 10000 samples (13% of tests with impact). By plotting several variables based on the scoring results for each test (in case of contact), we can visually see that there are 4 groups of data, which reinforces our assumption. In Figure 6-9 below, we can find a visualization of the headway, speed, deceleration, and speed of the target as functions of the risk assessment scores for each test.

The goal of the K-Means clustering algorithm is to group the samples according to their similarities and can be summarized in 4 steps [64] :

- 1) Randomly choose k centroids from the sample points as initial cluster centres.

- 2) Assign each sample to the nearest centroid.
- 3) Move the centroids to the centre of the samples that were assigned to it.
- 4) Repeat steps 2 and 3 until the cluster assignments do not change or the tolerance defined by user-or the maximum number of iterations is reached.

To measure the similarity between the samples, we define the squared Euclidean distance between two points x and y in m -dimensional space:

$$d(x, y)^2 = \sum_{j=1}^m (x_j - y_j)^2 = \|x - y\|_2^2 \quad (6.1)$$

Based on this metric, we can describe the K-Means algorithm as a simple optimization problem, an iterative approach that minimizes the Sum of Squared Errors (SSE) within each cluster.

$$SSE = \sum_{i=1}^n \sum_{j=1}^k w^{(i,j)} \|x^{(i)} - \mu^{(j)}\|_2^2 \quad (6.2)$$

Where $\mu^{(j)}$ is the centroid for cluster j , and $w^{(i,j)} = 1$ if the sample $x^{(i)}$ is in cluster j ; $w^{(i,j)} = 0$ otherwise.

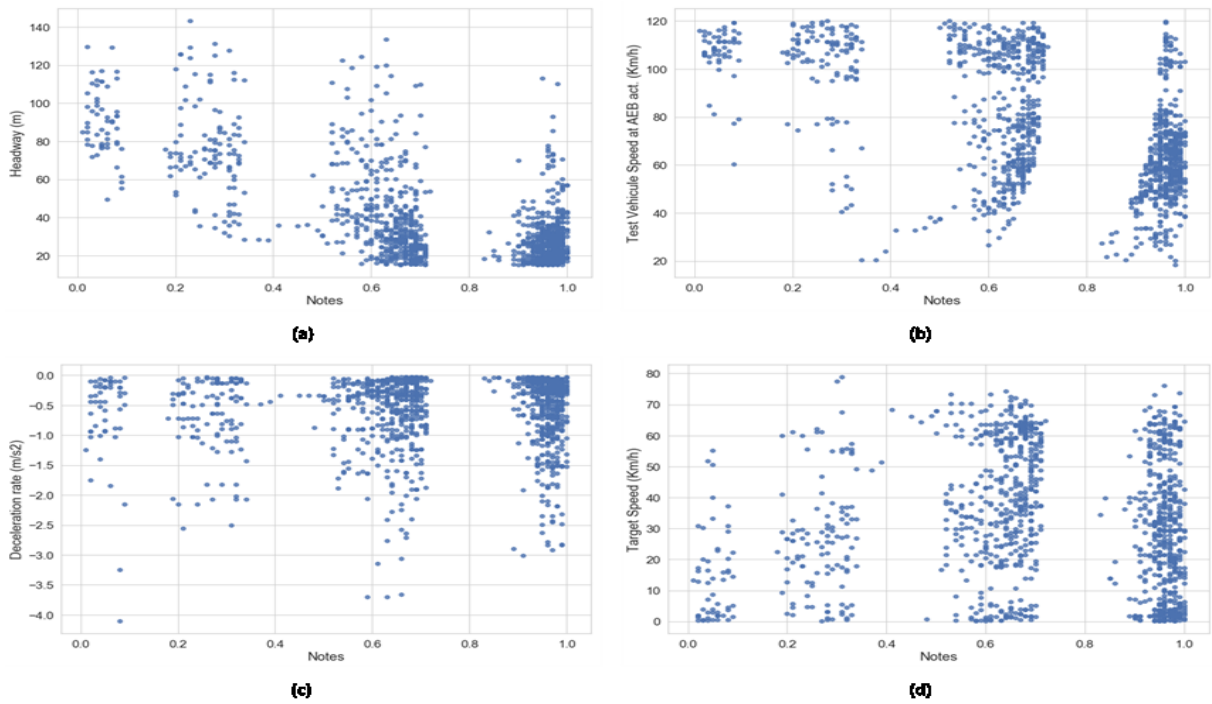


Figure 6-9 (a) Headway vs Notes - (b) Speed vs Notes - (c) Deceleration vs Notes - (d) Target speed vs Notes : [13% of tests with impact]

After applying the algorithm on our dataset, here are the clusters in Figure 6-10-(a) and (b), for headway and speed, that K-Means clustering has identified with the centroids of each cluster:

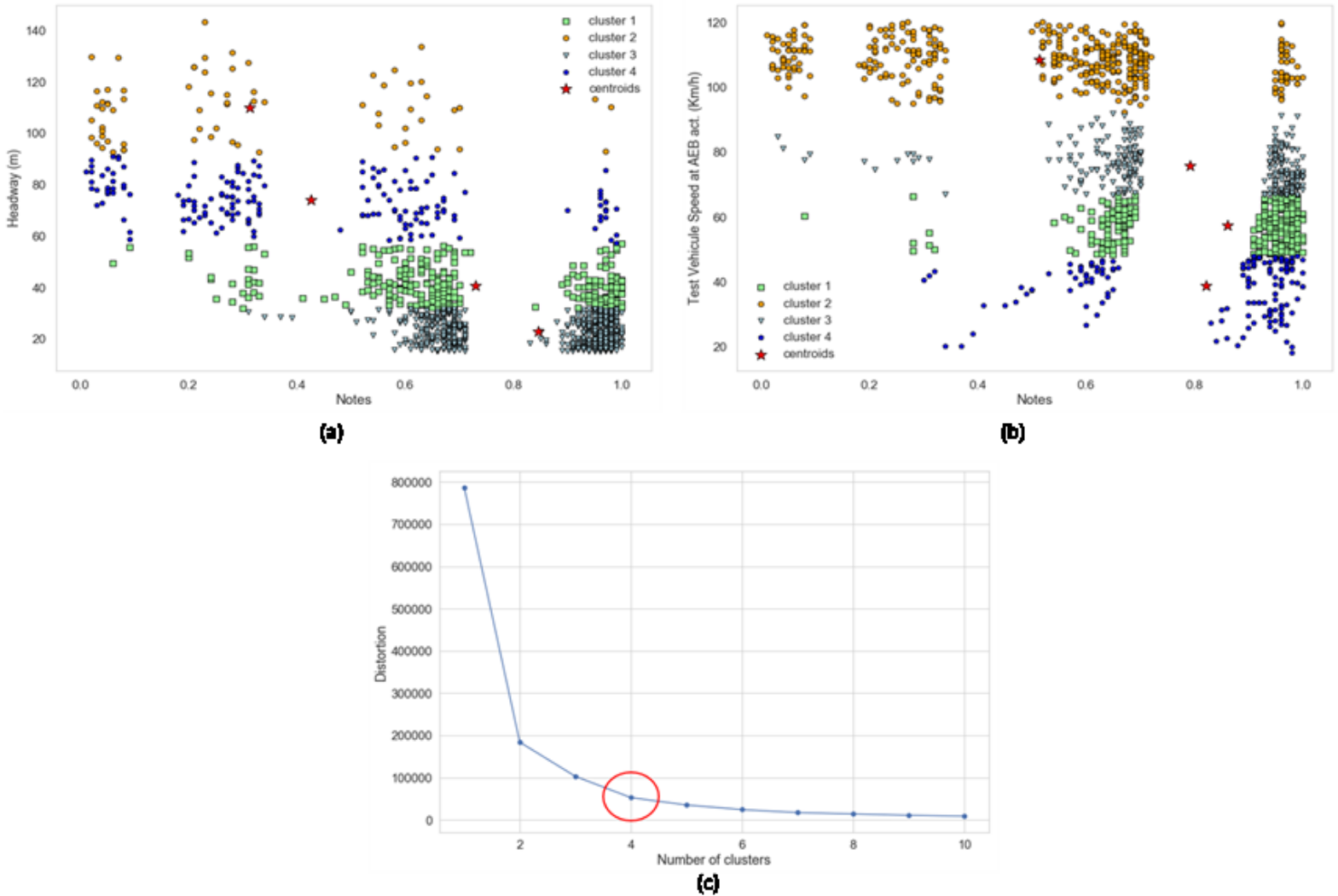


Figure 6-10 K-Means clustering results - 4 clusters - 10000 samples: (a) Headway - (b) Speed - (c) Elbow Method - Optimal Number of Clusters = 4

The same algorithm, with a different number of clusters (6 and 12) has been applied and the result remains unchanged in terms of grouping (clustering), the algorithm persists on doing a horizontal grouping, see figures in Appendix (Section 6.13.2).

To confirm the optimality of the selected cluster number, we applied the elbow method [190]. A graphical tool that can be used based on the cluster distortion SSE, to estimate the optimal

number of clusters k . The basic idea is to plot the distortion (SSE) for a range of values of k (here, from 1 to 10). We can say that, if k increases, the distortion will decrease and if the line chart looks like an arm, then the "elbow" on the arm is the value of k that is the best. As we can see in the following plot (Figure 6-10-(c)), the elbow is located at $k=4$, which is a good indication that $k=4$ is indeed the optimal choice for this dataset.

By analyzing the classification results above, we find that:

- Based on the logic of the rating system, a vertical grouping in relation to the level of risk is needed. This is because a test that has a score of 0.9 is less risky than one that has a score of 0.2 even if they have the same vehicle test speed. It is the impact speed reduction rate that comes into play.
- The number of tests with collision is insufficient, especially in regions with higher risk levels (between 0 and 0.8).

6.9.2 Case Study 2: 250000 Samples

For a better distinction between risk levels (horizontal and vertical), we have increased the number of samples used during the prediction stage to 250000 (13% of tests with impact). The idea is then to apply the same classification algorithm, to propose clear limits between risk levels (horizontal and vertical), so that we can finally retrieve and create a list of the worst-case scenarios for each risk zone.

Similar to the previous section, by plotting several variables based on the scores given for each test, we can visually see that there are 4 groups of data, which reinforces once again our assumption. In Figure 11-(a) and (b) below, we give a visualization of the headway and speed based on risk assessment scores for each test.

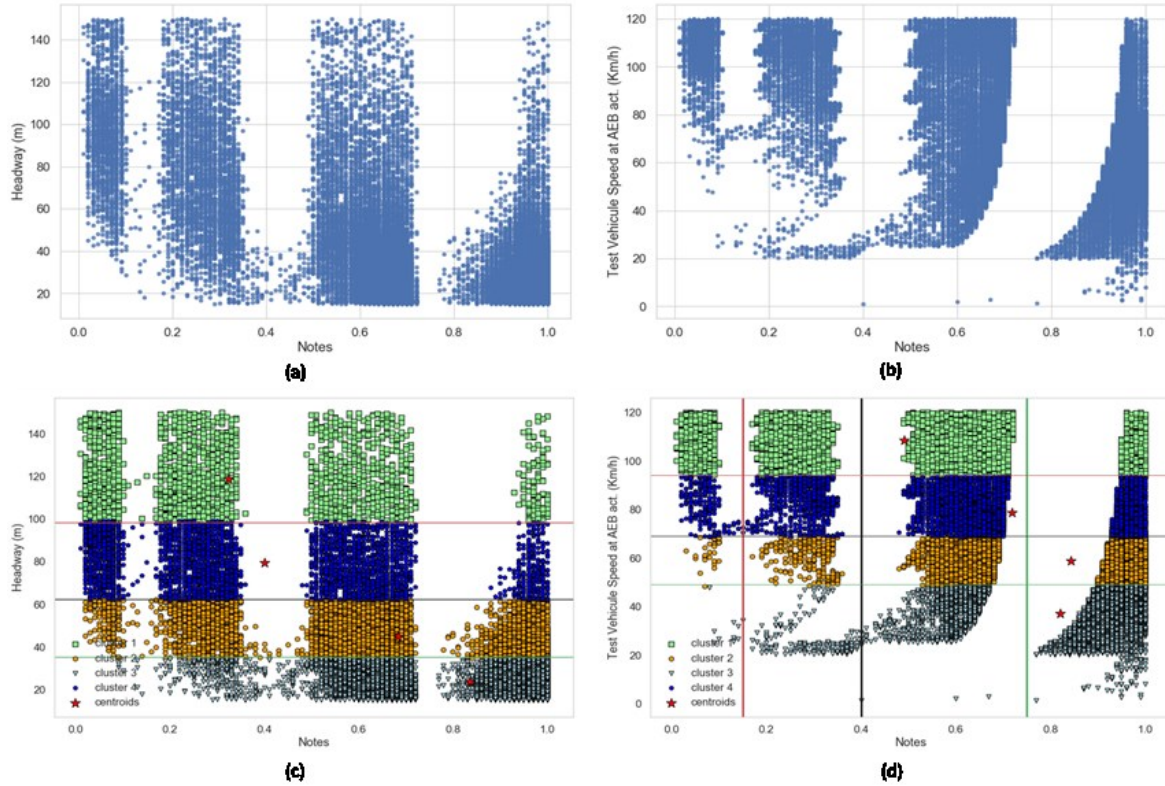


Figure 6-11(a) Headway vs Notes - (b) Speed vs Notes - (c) K-Means clustering results: Headway – and (d) Speed [250000 samples - 4 clusters]

After applying the algorithm on the new dataset, in Figure 11-(c) and (d) we can see the clusters that K-Means have identified with the centroids of each cluster.

Likewise, we can see in Figure 11-(c) that the K-Means algorithm has created 4 groups of data as expected, but with a horizontal grouping with respect to the distance levels. The same behaviour is observed with the speed of the testing vehicle (see Figure 11-(d)).

By analyzing the distribution of the notes, a vertical grouping of 4 risk levels can be proposed (see Figure 11-(d)). [Vertical lines → visual grouping and horizontal lines → K-Means clustering algorithm]:

1) Vertical classification : Vertical lines → visual grouping

- Level V1: $0 < Notes < 0.15$
- Level V2: $0.15 \leq Notes < 0.40$
- Level V3: $0.40 \leq Notes < 0.75$
- Level V4: $0.75 \leq Notes < 1$

2) Horizontal classification : Horizontal lines → K-Means grouping

- Level H1: $0 < Speed < 50$
- Level H2: $50 \leq Speed < 70$
- Level H3: $70 \leq Speed < 95$
- Level H4: $95 \leq Speed < 120$

This new proposal allows us to have both a better understanding and a better distinction between the levels of risks. We can now propose 16 risk zones, as mentioned in Figure 12, with "1" being the lowest and "16" being the highest.

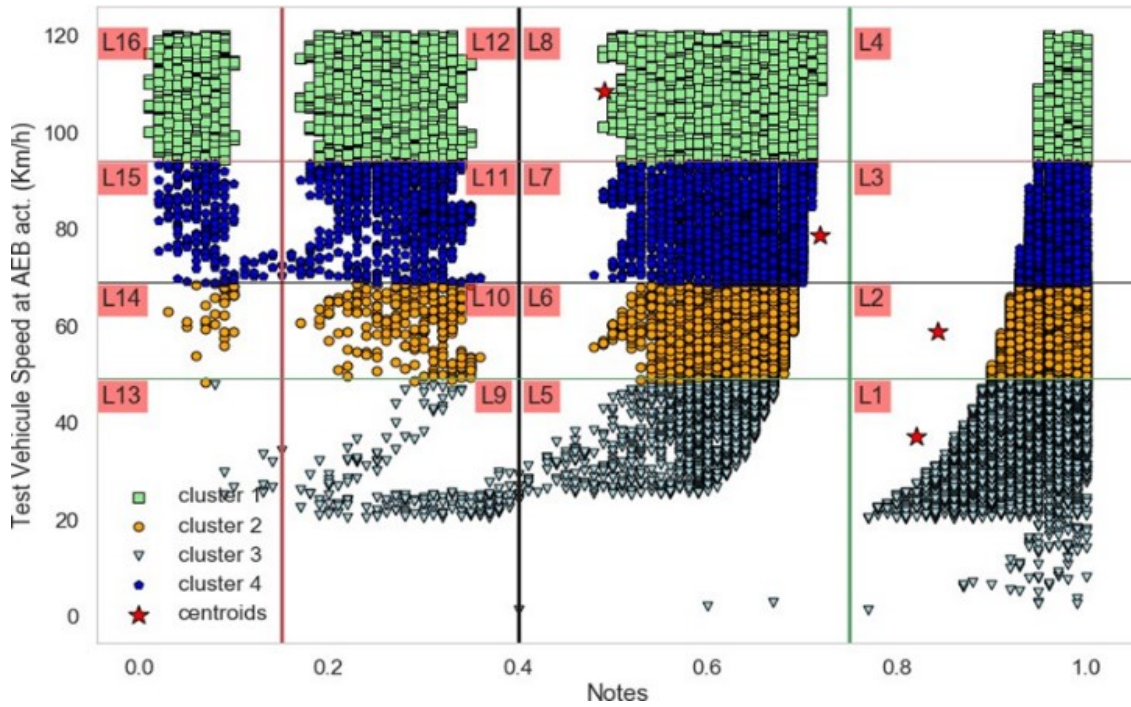


Figure 6-12 Risk classification: 16 risk zones

6.10 Final Selection [Worst-Case Scenarios]

The last step is simply to list the scenarios in each risk zone. This way we are taking the most challenging scenarios of each zone. With further analysis, we can eliminate the zones that are the least risky (L1 to L4) and call the rest of the zones (L5 to L16) as the most risky ones that contain the worst-case scenarios for this particular AEB system. It is clear that we cannot list them here one by one, for example zone 16 alone contains 1511 scenarios. Therefore, each risk zone will be represented by a statistical description of each parameter. In the sections below, we

take L16 as an example and the remaining zones, including the least risky ones (L1 to L4) can be found in Appendix. (Section 6.13.3).

6.10.1 Zone L16: Notes < 0.15 & Speed ≥ 95

Zone L16 is a bounded area that contains worst-case scenarios (1511 scenario) with test vehicle speeds higher than or equal to 95 km/h and a risk assessment scores (notes) less than 0.15. We begin with a histogram and the normal distribution of the speed, headway and target speed (see Figure 6-13-(a)). And for a better visualization of each input parameter distribution, here are below in Figure 6-13-(b) their representation in bar graphs.

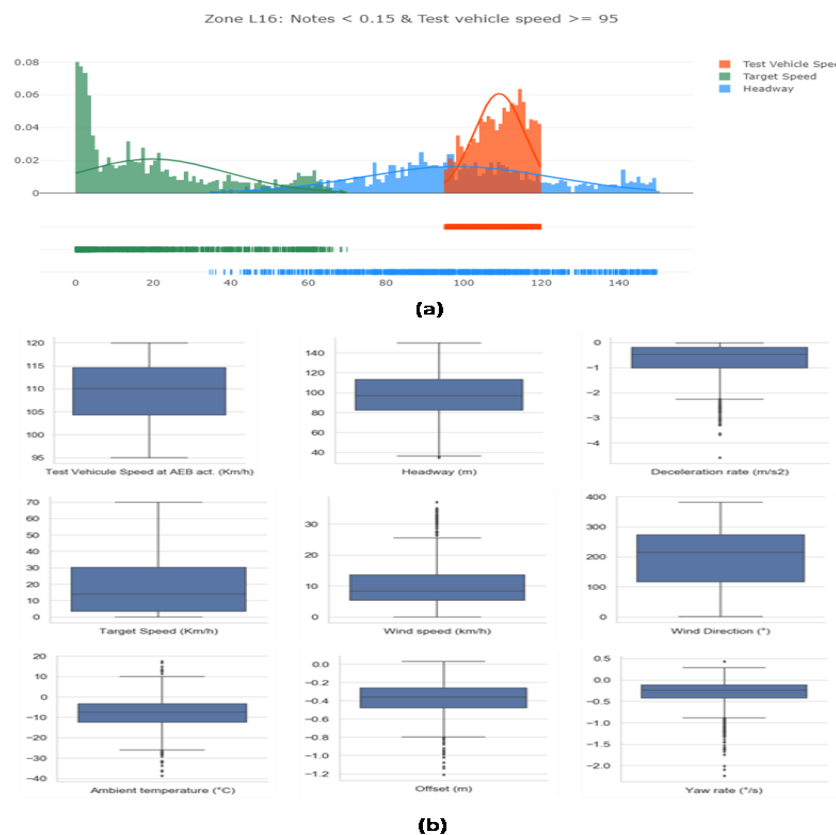


Figure 6-13 (a) ZoneL16 : Notes < 0.15 & Speed ≥ 95 - (b) Bar graph of each input parameter

Then in the Table 6-11 below, we generate a descriptive statistics of all the collected worst-case scenarios from the zone L16. These statistics summarize the central tendency, dispersion and shape of the distribution of the collected worst-case scenarios. The results include: count, mean, standard deviation, minimum, 25%, 50%, 75% percentiles and the maximum values of each parameter.

Table 6-11 Descriptive statistics [Zone L16]

	count	mean	std	min	25%	50%	75%	max
Ambient temperature (°C)	1511.00	-8.30	7.30	-38.73	-12.58	-7.63	-3.51	17.32
Wind speed (km/h)	1511.00	10.02	6.57	0.02	5.36	8.29	13.64	37.03
Wind Direction (°)	1511.00	198.56	92.22	1.11	116.68	213.84	272.98	381.91
Offset (m)	1511.00	-0.39	0.17	-1.21	-0.48	-0.36	-0.26	0.03
Yaw rate (°/s)	1511.00	-0.31	0.30	-2.24	-0.43	-0.24	-0.12	0.43
Headway (m)	1511.00	98.16	24.49	34.64	82.38	96.68	113.23	149.85
Deceleration rate (m/s ²)	1511.00	-0.73	0.70	-4.59	-1.03	-0.48	-0.20	-0.02
Test Vehicle Speed at AEB act. (km/h)	1511.00	109.19	6.57	95.00	104.22	110.00	114.56	119.97
Target Speed (km/h)	1511.00	19.79	19.13	0.00	3.30	13.94	30.30	69.94
Distance at warning (m)	1511.00	77.67	7.07	56.00	72.85	77.74	82.76	100.47
Distance at braking (m)	1511.00	30.86	7.42	13.78	25.76	30.41	35.21	59.14
Distance at AEB act.(m)	1511.00	16.08	8.07	-0.97	10.23	15.52	20.38	46.34
TTC at digital FCW (s)	1511.00	29.18	6.45	14.08	24.94	28.80	33.12	45.10
TTC at braking (s)	1511.00	4.37	0.59	3.03	3.94	4.29	4.74	6.33
TTC at AEB act.(m)	1511.00	-0.90	0.81	-2.59	-1.42	-0.91	-0.44	1.76
Speed at the impact (km/h)	1511.00	84.16	10.65	60.12	76.20	83.77	92.13	109.91
Shortest distance (m)	1511.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Reduction rate (%)	1511.00	22.92	8.60	5.08	15.98	23.30	30.26	38.73
Notes	1511.00	0.06	0.02	0.01	0.04	0.06	0.08	0.10

6.11 Web applications

Machine learning techniques are not only limited to offline analyzes and applications, they can also be deployed online as web applications. During our study, two web applications (Full-stack web applications) were developed and deployed online. Several tools and resources were used, here is a non-exhaustive list:

- Back-End: Python [Flask Framework, Dash Plotly Framework, Sklearn, JavaScript, etc.]
- Front-End: Dash (ReactJs), JavaScript, HTML, CSS
- Version control: Git, GitHub
- Hosting and deployment: <https://herokuapp.com> [191]

6.11.1 Web application 1

[URL: <https://liv-pmg-app1-nc.herokuapp.com> [192]] A web application that allows real-time predictions of the studied AEB system outputs for a given scenario. For example, if we take the scenario with the following inputs (1st column in Table 6-12), with a click of a button we get the following predicted results (2nd column). For the sake of comparison, we added here the real outputs for this specific scenario (3rd column). See the web app interface in Figure 6-14-(a) and (b).

Table 6-12 Inputs – Predicted outputs – Real outputs [web application 1]

Inputs	Predicted outputs (APP1)	Real outputs (Track test)
Test scenario : C1 Target type: SSV Ambient temperature : 2.16 °C Wind speed : 2.96 km/h Wind direction : 294.2 ° Offset : -0.075 m Yaw rate: -0.08 °/s Headway : 13.75 m Deceleration rate : -3.19 m/s ² Test speed : 55.02 km/h Target speed : 56.77 km/h	Distance at warning : 9.77 m Distance at braking : 7.52 m Distance at AEB act : 7.54 m TTC at digital FCW : 2.23 s TTC at braking : 1.33 s TTC at AEB act : 1.31 s Shortest distance : 0 m Speed at impact : 22.73 km/h Impact (yes/no) : Impact	Distance at warning : 10.7 m Distance at braking : 7.94 m Distance at AEB act : 7.84 m TTC at digital FCW : 2.5 s TTC at braking : 1.36 s TTC at AEB act : 1.33 s Shortest distance : 0 m Speed at impact : 21.58 km/h Impact (yes/no) : Impact

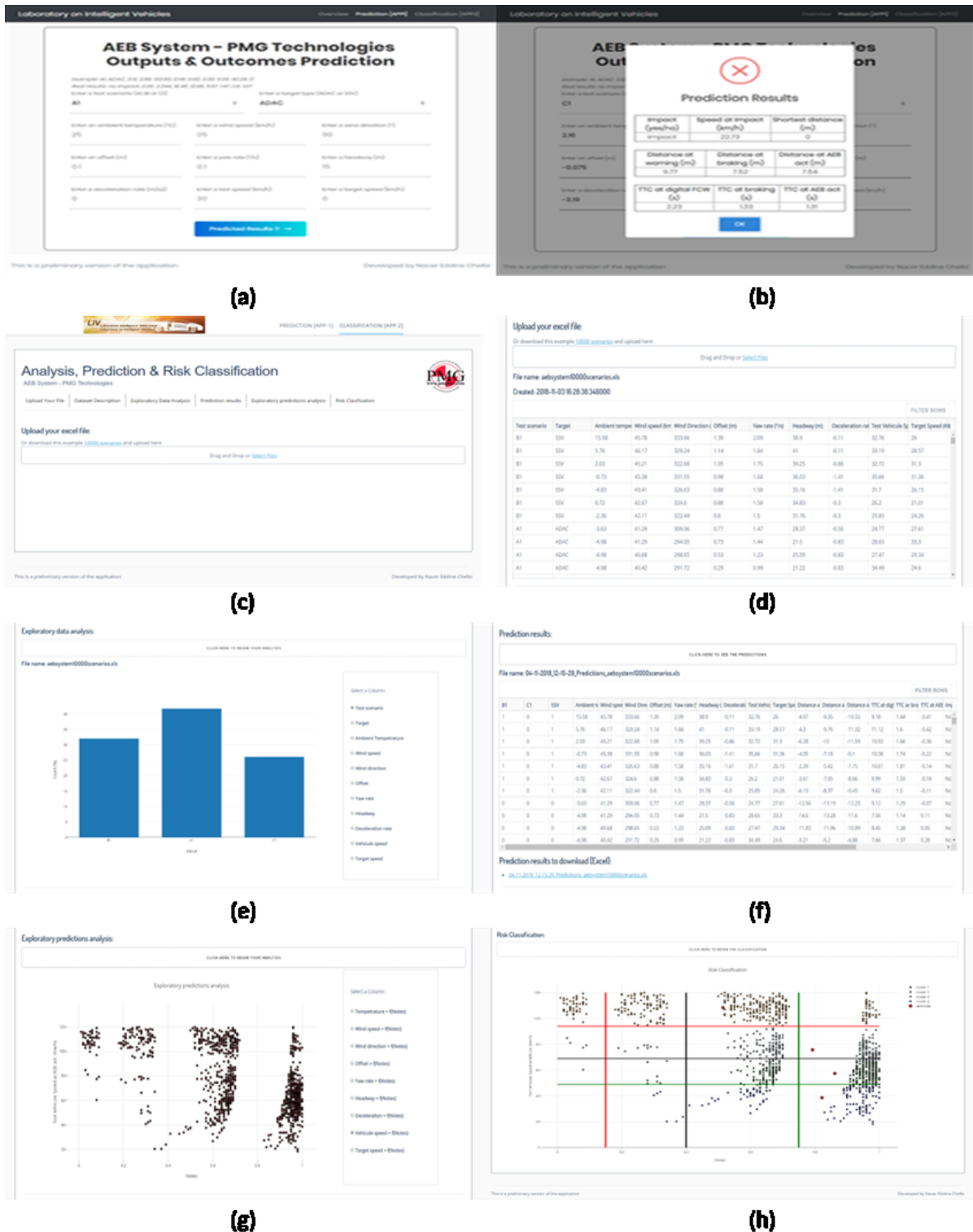


Figure 6-14 Web application 1 : (a) Interface – (b) Prediction results - Web application 2: (c) Interface – (d) Upload excel file and data description sections – (e) Exploratory data analysis - (f) Prediction results – (g) Exploratory prediction analysis – (h) Risk classification

6.11.2 Web application 2

[URL: <https://liv-pmg-app2-nc.herokuapp.com> [193]] An improved version of the first application that provides more options and features:

- The user can send an Excel file to the server containing the test scenarios
- Obtain a statistical description of the dataset sent to the server
- Real-time data analysis
- Real-time prediction results
- Real-time risk classification

In Figure 6-14-(c),(d),(e),(f),(g) and (h), we can see the web app interfaces.

6.12 Conclusion

In this chapter, we presented a worst-case scenario identification approach of preventive safety applications and advanced driver assistance systems based on machine and ensemble learning techniques. For starters, we used the physical test results of an AEB system to build its prediction models. For each output an appropriate machine or ensemble learning technique was used. The prediction results of the first 6 outputs: distance at warning, distance at braking, distance at AEB activation, TTC at digital FCW, TTC at braking and TTC at AEB activation, were satisfactory with accuracies that exceed 96% (between 96 and 99%). To improve the accuracy of the Impact (yes/no) output several machine and ensemble learning techniques were investigated and then used to produce a majority voting based classifiers. We noticed that the majority voting classifier performance outperform every other classifier with an AUC = 0.91 and an RMSE = 0.24, thus its selection as the main classifier (predictor) for the Impact (yes/no) output. For the prediction results of the last 2 outputs: speed at impact, shortest distance, we noticed that the prediction results of the speed at impact output are satisfactory with an accuracy = 95%, which is not the case for the shortest distance output with an accuracy = 73%. Since this output is predicted in case of non-collision, its prediction accuracy is not as important, i.e. when we don't have a collision it means that we are 100% safe and whether the vehicle stopped at 1 m or 0.73 m is not that important. In order to diagnose oversetting (high variance) or underfitting (high bias) of our models, the concept of Learning Curves was used. In our case, it allows us to decide whether an additional number of physical tests is necessary or not.

Next, a new sampling strategy based on the original, modified and multivariate Metropolis-Hastings algorithm (Component-Wide Updating) was applied to sample from the SPMD database. The sampling results showed the efficiency of the suggested sampling strategy with high acceptance rates and a very good coverage of the whole distributions. We continued by taking 10000 samples of each input parameter and provide them to the prediction model, then collect the predictions results (of each output), and finally, calculate the impact speed reduction rates. After that and to assess the risk of each test, in case of an impact, a proportional score to the reduction rate was attributed.

Then, for the risk classification, we applied the K-Means clustering technique and as a result, the algorithm has created 4 clusters of data (as expected) but with a horizontal grouping. To confirm the optimality of the selected cluster number, we applied the elbow method, and we found that $k=4$ is indeed the optimal choice for our dataset. However, and based on the logic of the rating system, a vertical grouping in relation to the level of risk was needed. Furthermore, the number of tests with collision is insufficient, especially in regions with higher risk levels. Next and for a better distinction between risk levels, we have increased the number of samples used during the prediction stage to 250000. We did apply the same classification technique and likewise we did get the same horizontal grouping but with a clear distinction between the risk levels. After that, by analyzing the distribution of the notes, a vertical grouping of 4 risk levels was proposed: The vertical lines are proposed based on a visual grouping and horizontal lines are proposed based on the K-Means clustering algorithm. This new proposal allows us to have both a better understanding and a better distinction between the levels of risks. We did propose 16 risk zones, with "1" being the lowest and "16" being the highest.

After that, the last step would be to list the scenarios of each zone and this way we took the most challenging scenarios of each zone. With further analysis, we did eliminate the zones that are the least risky (L1 to L4) and call the rest of the zones (L5 to L16) as the most risky ones which contain the worst-case scenarios for this particular AEB system. Then, each risk zone was represented by its statistical description that summarizes the central tendency, dispersion and shape of the distribution of the collected worst-case scenarios. The results include: count, mean, standard deviation, minimum, 25%, 50%, 75% percentiles and the maximum values of each parameter.

Finally, since the machine learning techniques are not only limited to offline analyzes and applications, they can also be deployed online as web applications, we presented two developed and deployed full stack web applications. The first one allows real-time predictions of the studied AEB system outputs for a given scenario. The second is an improved version of the first application that provides more options and features were: the user can send to an Excel file to the server containing the number of scenarios he wants, Obtain a statistical description of the dataset sent to the server, real-time data analysis, prediction results and risk classification.

In future work, we will focus on investigating more advanced clustering techniques for risk classification. We will also consider applying the proposed evaluation approach on different advanced driver assistance systems.

6.13 Appendix

6.13.1 Results: Sampling the FOT database

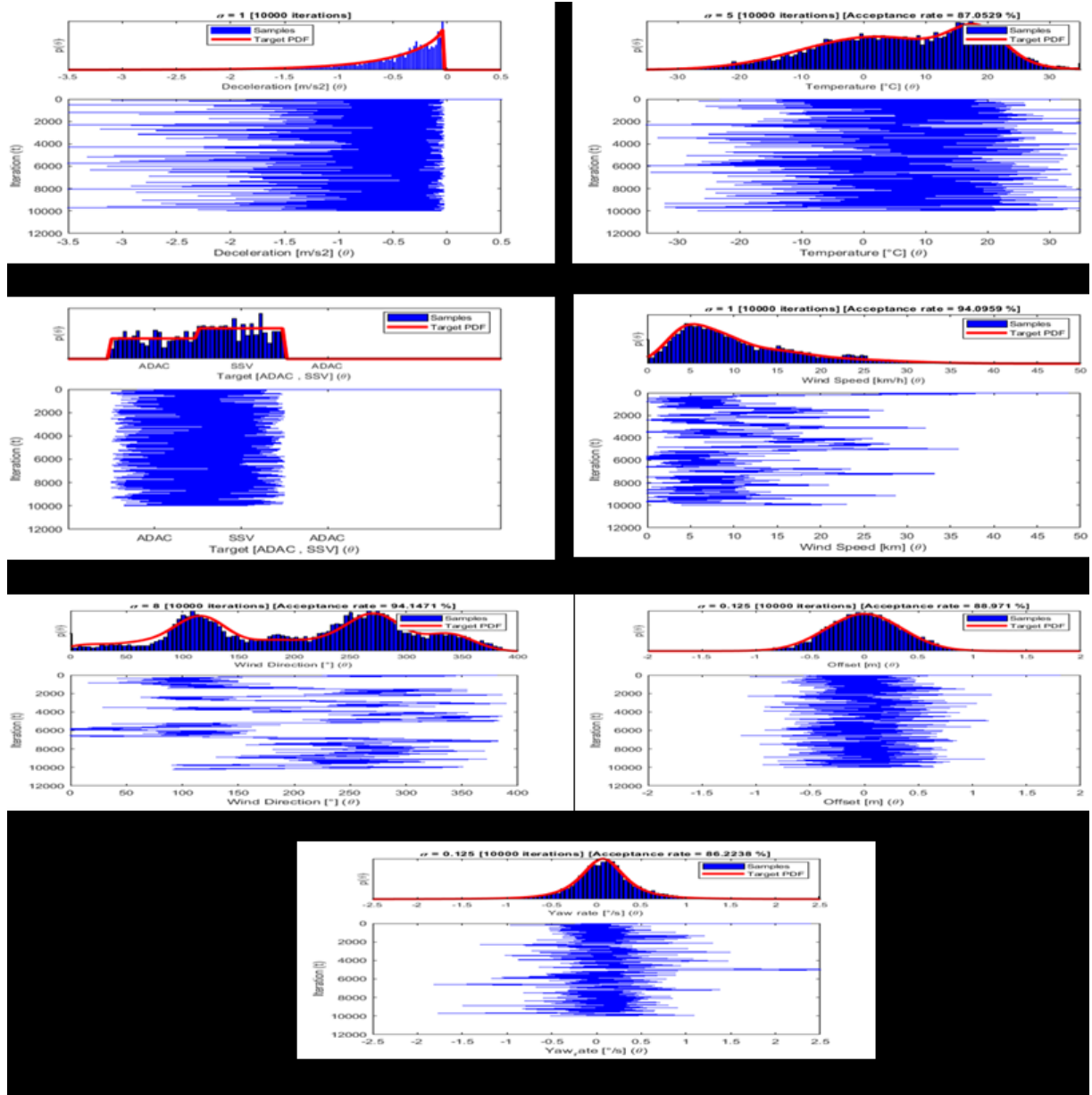


Figure 6-15 Sampling results - 10000 samples: (a) Deceleration - (b) Target - (c) Ambient temperature - (d) Wind speed - (e) Wind direction - (f) Offset - (g) Yaw rate

6.13.2 Results: Risk classification

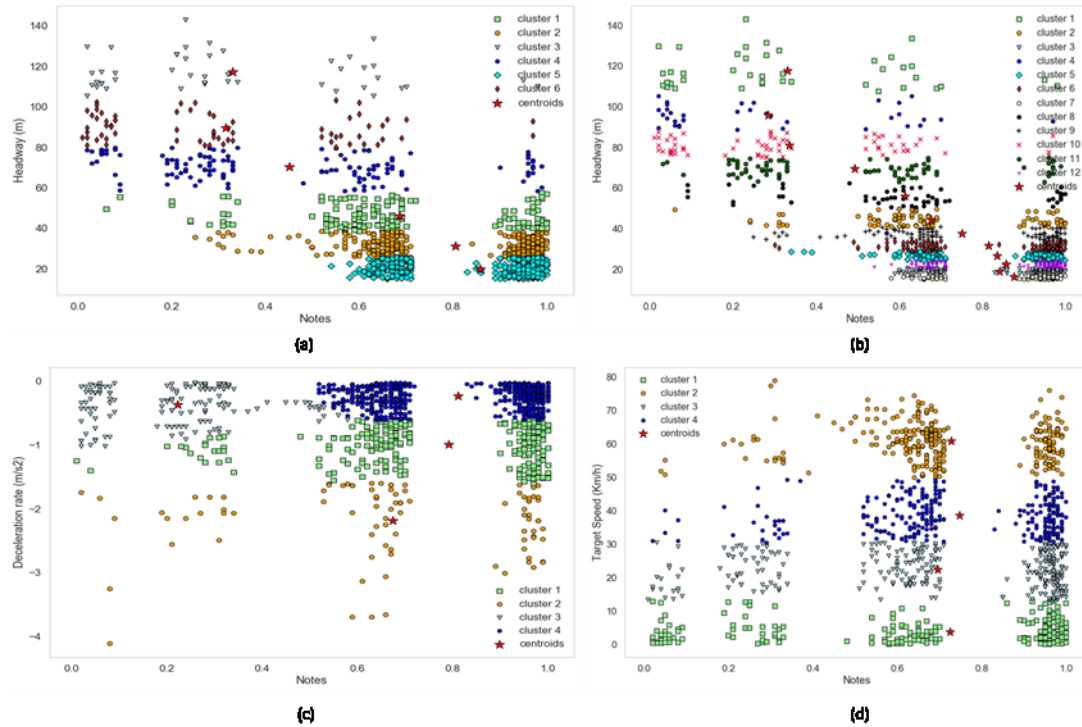


Figure 6-16 K-Means clustering - 10000 samples: (a) Distance - 6 clusters - (b) Distance - 12 clusters - (c) Deceleration - 4 clusters - (d) Target speed - 4 clusters

6.13.3 Results: Final selection [Worst-case scenarios]



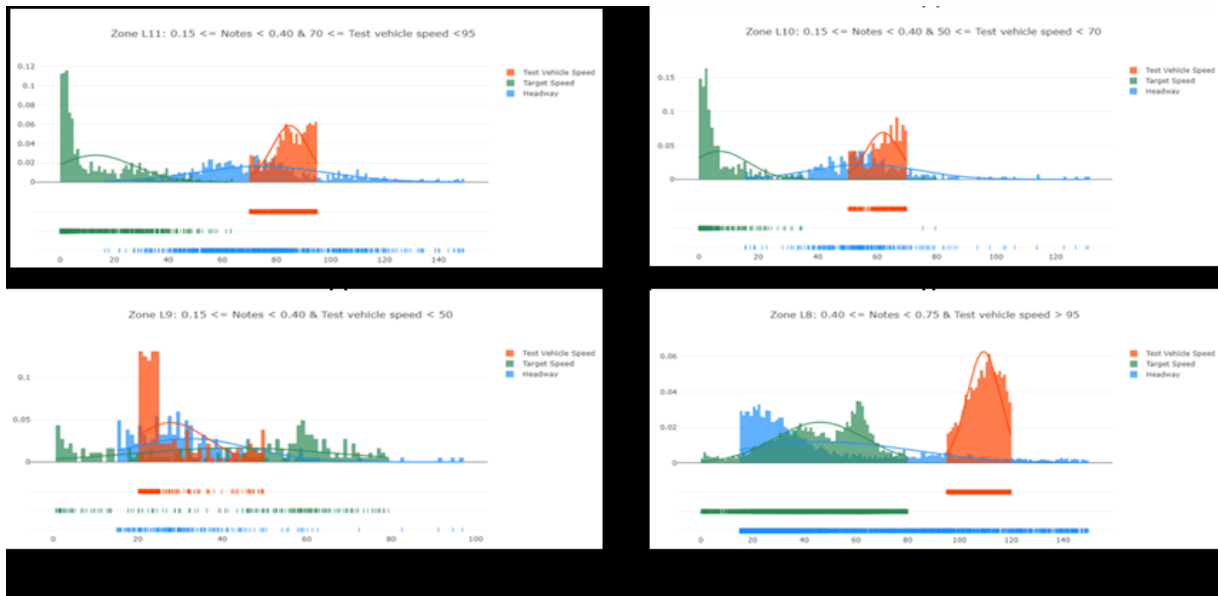


Figure 6-17 (a) Zone L15 - (b) Zone L14 - (c) Zone L13 - (d) Zone L12 - (e) Zone L11 - (f) Zone L10 - (g) Zone L9 - (h) Zone L8

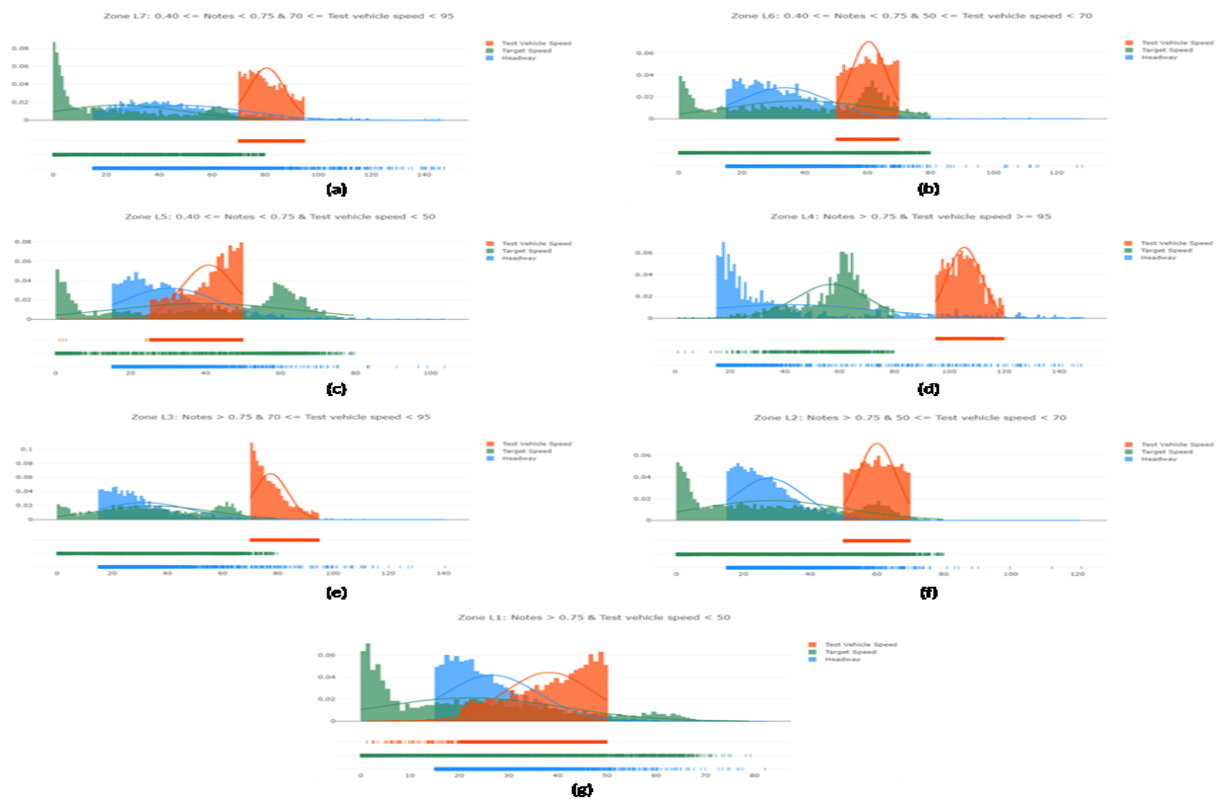


Figure 6-18 (a) Zone L7 - (b) Zone L6 - (c) Zone L5 - (d) Zone L4 - (e) Zone L3 - (f) Zone L2 - (g) Zone L1

Chapitre 7

Conclusion générale

Dans cette thèse de doctorat, nous avons proposé une approche d'«Évaluation Combinée» des systèmes avancés d'assistance et d'aide à la conduite dans des véhicules intelligents ou hautement automatisés sous de multiples conditions. L'idée de base consiste à identifier à la fois un ensemble de scénarios défavorables et de combiner les trois approches d'évaluations : simulations, essais sur piste et essais sur route, telle que suggérée par NHTSA.

Ce qui est unique à notre proposition est que l'approche est adaptée essentiellement dans un contexte où l'évaluation est conduite, pendant la phase de déploiement (ou de rappel), par une entité externe qui a un accès limité aux informations du système d'aide à la conduite à tester (c.-à-d. sa composition matériel ou logiciel et/ou son modèle mathématique et/ou ces entrées/sorties) et lorsque il est traité comme boîte noire. Nous avons proposé une nouvelle approche d'évaluation des systèmes avancés d'aide à la conduite dans un véhicule intelligent ou hautement automatisé – Approche d'« Évaluation Combinée ». Le concept « Évaluation Combinée », pendant la phase de déploiement, est nouveau dans le domaine des essais et d'évaluation de véhicules intelligents ou hautement automatisés. Il permet à la fois d'identifier un ensemble de scénarios défavorables et de combiner les trois approches d'évaluations, souvent utilisées séparément: simulations, essais sur piste et essais sur route.

7.1 Conclusions

Dans le chapitre 3, l'algorithme Metropolis-Hastings et le test de Kolmogorov ont été combinés pour développer une stratégie d'échantillonnage à partir d'une base de données de test opérationnel sur terrain (SPMD). Les résultats de simulation ont montré l'efficacité de la stratégie d'échantillonnage proposée avec des taux d'acceptation élevés et une très bonne couverture de l'ensemble des distributions. La qualité des résultats de la stratégie d'échantillonnage dépend principalement de la qualité de l'estimation des PDF et de l'optimisation de l'initialisation de l'algorithme de Metropolis-Hastings.

La version originale de l'échantillonneur de Metropolis-Hastings n'est applicable que sur des distributions continues, et pour l'appliquer sur des distributions discrètes, nous avons proposé dans le chapitre 4 une nouvelle version de l'algorithme, appelé : « Algorithme de Metropolis-Hastings modifié pour un échantillonnage efficace à partir de distributions de probabilités discrètes (MMH-DPD) ». Nous avons démontré, tout d'abord, la démarche à suivre pour remplacer la distribution discrète par un mélange de Gaussiennes généralisées et de comment affiner ces paramètres. Ensuite, une nouvelle étape après l'étape de rejet et d'acceptation est ajoutée, dans laquelle nous affectons chaque échantillon à son événement d'origine. Les résultats de simulations ont montré l'efficacité de la nouvelle version de l'algorithme « MMH-DPD » avec des erreurs qui tendent pratiquement vers zéro. L'algorithme crée une chaîne de Markov qui génère des échantillons avec une très grande précision de telle sorte que lorsque le nombre d'échantillons tend vers l'infini, on peut garantir que la nouvelle distribution de probabilité des échantillons récupérés à l'aide de l'algorithme MMH-DPD tend vers la vraie distribution discrète.

Dans le chapitre 5, nous avons proposé une approche d'évaluation virtuelle des applications de sécurité préventive et de fonctions avancées d'aide à la conduite. L'objectif consistait à identifier les pires scénarios pour une application ADAS donnée, l'AEB dans cette thèse, basée sur l'analyse de données FOT et la simulation de différents scénarios de tests sur une plateforme de simulation véhiculaire. Avec cent tests virtuels, nous avons pu démontrer comment extraire les pires des cas à risques en se basant sur une évaluation et une classification non supervisées des risques. L'inclusion d'une stratégie d'échantillonnage basée sur l'algorithme Metropolis-Hastings, nous a permis de créer une nouvelle stratégie d'échantillonnage facile à mettre en œuvre, robuste, et qui garantit quasiment une convergence vers la distribution exacte et, enfin, une approche globale avec une couverture pondérée sur tout l'espace opérationnel. La qualité de nos résultats dépend essentiellement de trois facteurs :

- 1) La qualité de la stratégie d'échantillonnage appliquée; qui dépend elle-même de la qualité de l'estimation des PDF et de l'optimisation de l'initialisation de l'algorithme de Metropolis-Hastings.
- 2) La fiabilité du logiciel de simulation véhiculaire à reproduire les scénarios de tests réels.

- 3) Les spécifications d'évaluations; qui comprend la description du système et de ces fonctions, les objectifs d'évaluation et les indicateurs clés, commune pour toutes les évaluations (facteurs techniques, humains, potentiels de sécurité).

Dans le chapitre 6, nous avons présenté une approche d'identification des pires scénarios de tests réels, pour des applications de sécurité préventive et de systèmes avancés d'aide à la conduite, basées sur des techniques d'apprentissages automatiques et ensemblistes. Pour démontrer notre approche, nous avons utilisé les résultats des tests physiques d'un système AEB mené chez PMG Technologies. Pour chaque sortie du système et pour élaborer son modèle de prédiction, une technique d'apprentissage automatique ou ensembliste appropriée a été utilisée. Les résultats de prédictions sont satisfaisants avec des précisions qui dépassent 95%. Ensuite, une nouvelle stratégie d'échantillonnage basée sur l'algorithme original, modifié et multivarié de Metropolis-Hastings (Component-Wide Updating) a été appliquée pour échantillonner à partir de la base de données FOT (SPMD). Les résultats de l'échantillonnage ont montré l'efficacité de la stratégie d'échantillonnage proposée avec des taux d'acceptation élevés et une très bonne couverture de l'ensemble des distributions. Nous avons ensuite procédé à l'évaluation et la classification de chaque test. Pour la classification des risques, nous avons appliqué une technique d'apprentissage automatique non supervisé qui nous a permis de proposer des niveaux et des zones de risques. Après cela, la dernière étape consisterait à répertorier les scénarios de chaque zone pour éliminer ensuite les zones les moins risquées et garder les zones les plus risquées contenant les pires scénarios. Finalement et pour une analyse en ligne et en temps réel, nous avons développé et déployé deux applications web (Full-Stack) sur un serveur externe (Heroku). La première permet de prédire en temps réel les sorties du système AEB étudié pour un scénario donné. La seconde offre plus d'options et de fonctionnalités où l'utilisateur peut envoyer au serveur un fichier Excel contenant le nombre de scénarios qu'il souhaite tester.

Puisque l'approche d'évaluation proposée combine les trois approches d'évaluations telles que mentionnées dans la Figure 7-1, les résultats obtenus montrent qu'il y a un grand potentiel de réduire le temps d'évaluation et le nombre d'essais physiques des systèmes avancés d'assistance et d'aide à la conduite pendant la phase de déploiement d'une manière considérable.

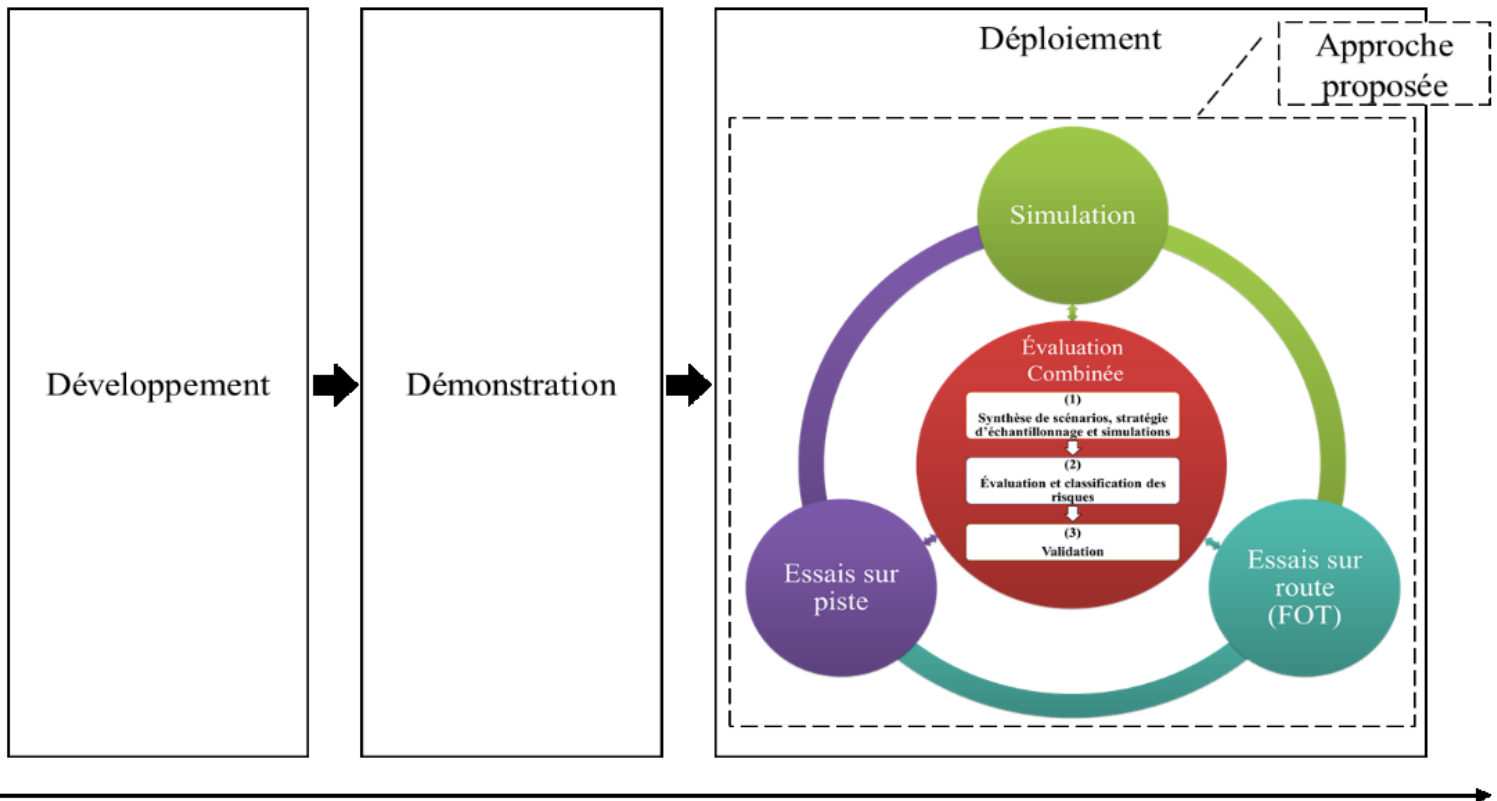


Figure 7-1 Stades de développement, de démonstration et de déploiement

7.2 Limitations et perspectives de recherche

Bien que nous avons appliqué avec succès l'approche d'évaluation proposée sur un système AEB, nous pouvons faire davantage pour adapter, améliorer l'efficacité et d'appliquer l'approche proposée sur d'autres systèmes ADAS tels que : ACC, LKA, etc. Dans cette partie, nous discutons quelques perspectives pour les futurs travaux. Nos suggestions sont classées comme suit :

7.2.1 Intégration d'une solution HIL ou VeHIL ou ViL

Une solution HIL est souvent utilisée pendant la phase de développement des systèmes chez les constructeurs. Cependant et avec l'aide des techniques de reverse engineering, une telle solution peut être envisagée et intégrée pendant la phase de déploiement chez des entités tierces (voir Annexe B). Une autre bonne alternative serait le VeHIL. Tel que discuté dans l'état de l'art, la solution consiste en l'intégration d'un véhicule dans la boucle de validation, mais ça reste dans

un environnement de laboratoire. Le principe de fonctionnement de cette méthode de validation est mentionné dans la Figure 7-2. Dans un laboratoire VeHIL, un environnement virtuel est défini dans lequel le véhicule, l'infrastructure et leurs interactions sont simulés en temps réel. Une deuxième configuration ViL (Vehicle-in-the-Loop) a été développée par TASS International [65] (voir Figure 7-2). Elle est destinée aux systèmes Adaptive Cruise Control (ACC) et Crash Imminent Braking (CIB) mais peut également être adaptée pour d'autres applications ADAS. Cette solution supporte la vérification du véhicule complet de manière sûre et contrôlée en injectant des données de capteurs virtuels dans le véhicule.



Figure 7-2 Test des applications ADAS avec la solution ViL [65]

7.2.2 Amélioration de la stratégie d'échantillonnage

Malgré les puissances de calcul que nous disposant aujourd'hui, les algorithmes MCMC restent très coûteux en termes de calculs. Plusieurs milliers d'itérations sont nécessaires pour obtenir des estimations à faible variance, ce qui représente un inconvénient majeur. Plusieurs approches ont été proposées qui traitent et expliquent comment on réduit le coût et le temps de calcul des algorithmes MCMC, nous citons : «Multiple Try Metropolis», «Ensemble MCMC» et «Prefetching Approaches». Une autre idée consiste à paralléliser le calcul des différentes chaînes sur plusieurs machines. Un calcul parallèle diminue le temps de calcul, mais nécessite des ressources matérielles supplémentaires. Les méthodes MCMC sont beaucoup plus performantes dans des dimensions plus élevées, nous l'avons prouvé avec deux dimensions. Par contre, il serait intéressant de passer à des dimensions supérieures.

7.2.3 Application des techniques de reverse engineering pour l'identification des systèmes ADAS

En ce qui concerne l'identification des systèmes ADAS, deux solutions sont envisageables, soit:

- 1- Chaque constructeur fournit un modèle mathématique (ou un modèle de simulation) de son système. La validation du modèle doit être faite par les soins de l'entité tierce.
- 2- Ou bien, une procédure d'identification sera conduite en interne (par les soins de l'entité tierce) pour modéliser le système.

L'objectif principal est de choisir un système d'aide à la conduite, enregistrer les entrées et les sorties/comportement/réactions résultantes d'un essai réel, pour arriver finalement à l'identifier le plus fiable possible. Ce modèle sera par la suite la base de toute validation de simulation. Plusieurs approches peuvent être appliquées. Dans cette étude, nous avons appliqué une approche basée sur des techniques d'apprentissage automatique. Souvent, une procédure d'identification commence par un protocole d'expérimentation, un choix de la structure du modèle, choisir des critères de décisions, une étape d'estimation du modèle et finalement l'étape de validation. Si l'étape de validation ne donne pas les résultats attendus, il faut récupérer un autre jeu de données et réviser le choix de la structure du modèle et du critère de décision. En appliquant des techniques de reverse engineering, il est primordial de conserver l'intégrité du système.

7.2.4 Application de techniques avancées de classification et de clustering

Cette partie peut être abordée par deux approches :

- Application de techniques avancées de classification par apprentissage non supervisé (sans étiquetage à l'avance), nous citons : Regroupement hiérarchique, DBSCAN, OPTICS, les réseaux de neurones, etc.
- Application de techniques avancées de classification par apprentissage supervisé. Pour avoir un étiquetage à l'avance, il serait intéressant d'appliquer la méthode ASIL pour classer les risques : Il existe quatre niveaux ASIL identifiés par la norme ISO26262 : ASIL A (étant le plus faible), ASIL B, ASIL C, ASIL D (étant le plus élevé).

Annexe A

Méthodes d'apprentissage automatique et ensembliste pour l'évaluation des véhicules intelligents/autonomes

A.1 Introduction

A.1.1 Méthodes de régression

Les modèles de régression sont souvent utilisés pour prédire des valeurs réelles telles qu'une distance de freinage ou une vitesse d'impact, par exemple. Si notre variable indépendante est le temps, on parle de prévisions de valeurs futures, sinon notre modèle prédit des valeurs actuelles, mais qui sont inconnues. Dans cette partie, nous allons présenter les méthodes de régressions suivantes :

- Régression linéaire simple
- Régression linéaire multiple (MLR : Multiple Linear Regression)
- Régression à vecteurs de support (SVR : Support Vector Regression)
- Régression par processus gaussien (GPR : Gaussian Process Regression)

A.1.1.1 Régression linéaire simple

La régression linéaire simple est une technique d'apprentissage automatique supervisé qui permet la prédiction d'une réponse quantitative à l'aide d'une seule variable d'entrée ("caractéristique" ou "variable explicative"). Plus simplement, cette technique essaye d'ajuster une ligne droite à un ensemble d'observations, pour prédire ensuite une variable dépendante (y) en fonction des valeurs d'une variable indépendante (x). Il est supposé que les deux variables sont linéairement dépendantes et son modèle est de la forme suivante [43]:

$$y = b_0 + b_1x \tag{7.1}$$

Ou b_0 est le terme d'interception, c'est-à-dire la valeur attendue de y lorsque $x = 0$, et b_1 est la pente, l'augmentation de y associée à une unité d'augmentation de x . Généralement, ces coefficients sont estimés en minimisant un critère des moindres carrés. Et pour le faire, soit :

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \quad (7.2)$$

représentent n paires d'observations, chacune consistant en une mesure de x et une mesure de y . L'objectif est d'obtenir des estimations des coefficients \hat{b}_0 et \hat{b}_1 telle que le modèle linéaire (0.1) correspond bien aux observations, c'est-à-dire que $y_i = \hat{b}_0 + \hat{b}_1 x_i$ pour $i = 1, \dots, n$, donc la ligne résultante est aussi proche que possible des n points d'observations.

Soit $\hat{y}_i = \hat{b}_0 + \hat{b}_1 x_i$ la prédiction de y basée sur la i ème valeur de x . Alors $e_i = y_i - \hat{y}_i$ représente le i ème résidu, c'est la différence entre la valeur de réponse observée et la valeur de réponse prédite par notre modèle linéaire. Nous définissons la somme résiduelle des carrés (*RSS* : *Residual Sum of Squares*) comme:

$$RSS = e_1^2 + e_2^2 + \dots + e_n^2 \quad (7.3)$$

ou équivalent à :

$$RSS = (y_1 - \hat{b}_0 - \hat{b}_1 x_1)^2 + (y_2 - \hat{b}_0 - \hat{b}_1 x_2)^2 + \dots + (y_n - \hat{b}_0 - \hat{b}_1 x_n)^2 \quad (7.4)$$

La méthode des moindres carrés choisit \hat{b}_0 et \hat{b}_1 qui minimise le *RSS*. On peut montrer que les coefficients qui minimisent l'équation (0.4) sont :

$$\begin{aligned} \hat{b}_0 &= \bar{y} - \hat{b}_1 \bar{x} \\ \hat{b}_1 &= \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \end{aligned} \quad (7.5)$$

où $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ et $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ sont les moyennes des observations (échantillon). En d'autres termes, l'équation (0.5) définit les estimations des coefficients des moindres carrés pour la régression linéaire simple.

A.1.1.2 Régression linéaire multiple (MLR)

La régression linéaire simple peut facilement être étendue pour inclure plusieurs variables d'entrées. C'est ce qu'on appelle la régression linéaire multiple. Cette dernière tente de modéliser la relation entre deux variables ou plus en ajustant une équation linéaire aux données observées. La forme générale du modèle d'une régression linéaire multiple est :

$$y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_kx_k - \varepsilon \quad (7.6)$$

Où b_0, b_1, b_2, \dots représente les coefficients de régression, x_0, x_1, x_2, \dots sont les variables indépendantes, y est la variable dépendante et ε est l'erreur aléatoire reflétant la différence entre la relation linéaire observée et prédite.

Soit une expérience effectuée n fois, si nous obtenons n groupe d'observations $x_{i1}, x_{i2}, \dots, x_{in}$, alors le modèle de régression linéaire peut être exprimé par [45]:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_k \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix} \quad (7.7)$$

En général, le modèle avec k variables d'entrées (explicatives) peut être exprimé par :

$$y = X\beta + \varepsilon \quad (7.8)$$

Où $y = (y_1, y_2, \dots, y_n)^T$ est un vecteur $n \times 1$ de n observations sur une variable d'étude,

$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nk} \end{pmatrix}$ est une matrice $n \times k$ de n observations sur chacune des k variables

explicatives, $\beta = (b_1, b_2, \dots, b_k)^T$ est un vecteur $k \times 1$ des coefficients de régression et

$\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)^T$ est un vecteur $n \times 1$ de composantes d'erreurs aléatoires (ou termes de

perturbations. Si le terme d'interception est présent, la première colonne de X sera $(1, 1, \dots, 1)^T$.

La procédure d'estimation du vecteur des coefficients de régression consiste à minimiser :

$$\sum_{i=1}^n Q(\varepsilon_i) = \sum_{i=1}^n Q(y_i - x_{i1}b_1 - x_{i2}b_2 - \dots) \quad (7.9)$$

Pour une fonction Q convenablement choisie.

Nous considérons le principe des moindres carrés ordinaire (OLS : Ordinary Least Squares).

Soit B l'ensemble des vecteurs possibles β . L'objectif consiste à trouver un vecteur

$b_{\min} = (b_1, b_2, \dots, b_k)^T$ dans B qui minimise la somme des carrés des déviations ε_i , c'est-à-dire :

$$S(\beta) = \sum_{i=1}^n \varepsilon_i^2 = \varepsilon' \varepsilon = (y - X\beta)'(y - X\beta) \quad (7.10)$$

En dérivant $S(\beta)$ par rapport à β et le mettre égale à zéro $\left(\frac{\partial S(\beta)}{\partial \beta} = 0 \right)$, on peut montrer que

coefficients qui minimisent l'équation (0.10) sont :

$$b_{\min} = (X'X)^{-1} X'y \quad (7.11)$$

Qui est appelé estimateur des moindres carrés ordinaires (OLSE: Ordinary Least Square Estimator) de β .

A.1.1.3 Régression à vecteur de support (SVR)

La régression à vecteurs de support (SVR) est une généralisation de la méthode de classification de machine à vecteur de support (SVM : Support Vector Machine) [voir section], dans laquelle le modèle prédit une sortie à valeur continue, par opposition à une sortie d'un ensemble fini [46]. La technique a été introduite pour la première fois en 1995 par Vapnik [47]. Vapnik a proposé une SVM appelée ε -support vector regression (ε -SVR), qui effectue des tâches de prédiction à partir de la fonction de perte ε -intensive [46]-[48].

Soit $x_i \in \mathbf{R}^m, i = 1, \dots, N$ un ensemble d'observations de l'entrée (variable indépendante) d'entrée, et soit $t_i \in \mathbf{R}$ un ensemble d'observations de la sortie cible (variable dépendante). La fonction à approximer peut-être écrite comme suit :

$$t_i \approx y_i = w^T x_i + b \quad (7.12)$$

Dans laquelle, on suppose que y_i sont les valeurs de prédiction, w est la matrice des coefficients et b représente le biais. La formulation du problème de SVR est souvent mieux comprise d'un point de vue géométrique, voici un exemple unidimensionnel dans la Figure 7-3.

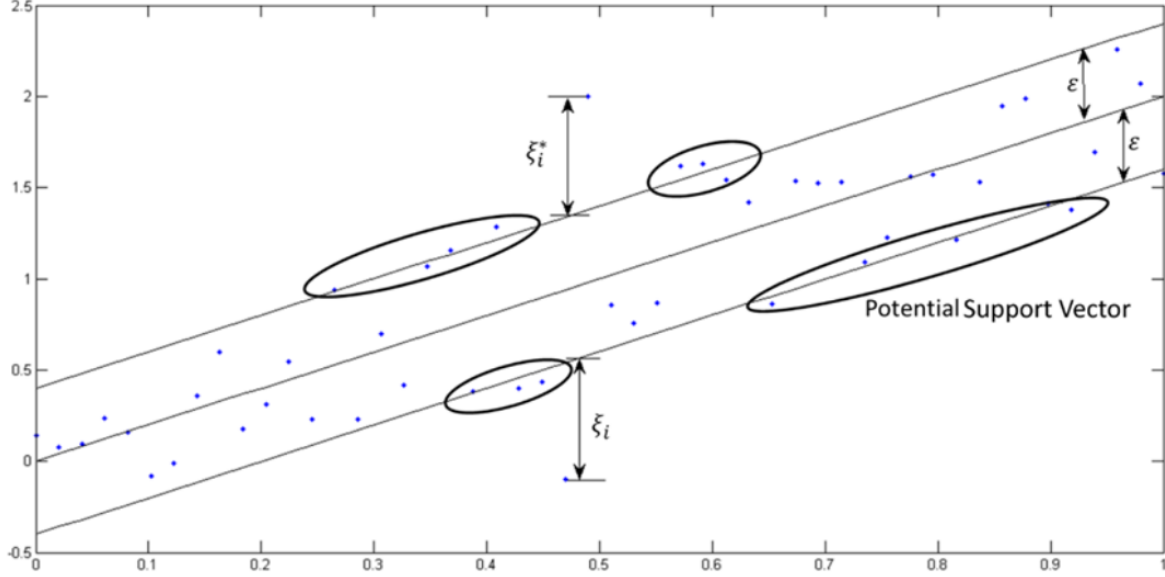


Figure 7-3 SVR linéaire à une dimension [r4]

Bien que les données d'entrées doivent rester dans une tolérance d'erreur fiable (ϵ), l'erreur peut être considérée négligeable tant que ça reste dans la plage $\pm\epsilon$. Dans le cas contraire, Vapnik et al [47] ont proposé la fonction de perte suivante :

$$L_{\epsilon}(t_i, y_i) = \begin{cases} 0 & |t_i - y_i| \leq \epsilon \\ + & \text{Sinon} \end{cases} \quad (7.13)$$

Où (ϵ) est un paramètre libre qui sert comme un seuil. Si $t_i - y_i$ se situe hors de la plage $\pm\epsilon$, l'erreur serait égal à ξ_i :

$$\xi_i = |t_i - y_i| - \epsilon \quad (7.14)$$

Par conséquent, la fonction de perte peut être considérée comme suit :

$$L_{\varepsilon}(t_i, y_i) = \begin{cases} 0 & |t_i - y_i| \leq \varepsilon \\ |t_i - y_i| - \varepsilon & \text{otherwise} \end{cases} \quad (7.15)$$

Avec ces suppositions, le processus de régression peut être transformé en un problème de minimisation et le problème d'optimisation peut être écrit comme suit [46]-[49]:

$$\begin{aligned} \min & \frac{1}{2} w^T w + C \sum_{i=1}^N (\xi_i^+ + \xi_i^-) \\ \text{s.à} & \\ & -t_i + y_i + \varepsilon + \xi_i^+ \geq 0 \\ & t_i - y_i + \varepsilon + \xi_i^- \geq 0 \\ & \xi_i^+ \geq 0 \\ & \xi_i^- \geq 0 \end{aligned} \quad (7.16)$$

Où ξ_i^+ et ξ_i^- sont les termes de pénalités et le paramètre $C > 0$ qui est utilisé comme paramètre de régularisation, c'est-à-dire, un paramètre ajustable. Par exemple, un plus grand C favorise la minimisation de l'erreur.

Les multiplicateurs de Lagrange seront utilisés pour résoudre la double forme du problème d'optimisation, pour avoir finalement une forme linéaire du modèle SVR :

$$y = \sum_i (\alpha_i^+ - \alpha_i^-) x_i^T x_j + b \quad (7.17)$$

Où α_i^+ et α_i^- sont les multiplicateurs de Lagrange. La forme non linéaire du modèle SVR est comme suit :

$$y = \sum_i (\alpha_i^+ - \alpha_i^-) K(x_i, x_j) + b \quad (7.18)$$

Un noyau Gaussien est souvent utilisé $K(x_i, x_j) = \exp\left(-\frac{1}{2\sigma^2} \|x_i - x_j\|^2\right)$ pour construire le modèle SVR non linéaire.

A.1.1.4 Régression par processus gaussien (GPR)

Les modèles de régression par processus gaussien (GPR) sont des modèles probabilistes flexibles, non linéaires et non paramétriques où la fonction $f(x)$, à estimer, est représentée par un processus Gaussien. Considérons l'ensemble d'observations (d'entraînement) $\{(x_i, y_i); i=1, 2, \dots, n\}$, where $x_i \in \mathbf{R}^m$ and $y_i \in \mathbf{R}$ et le modèle de régression non linéaire suivant [50]-[53]:

$$y = f(x) + \varepsilon \quad (7.19)$$

Où $f(\cdot)$ est la fonction à estimer. Par la méthode de régression par processus gaussien, $f(\cdot)$ est supposé suivre un processus gaussien avec une fonction moyenne $\mu(\cdot)$ et une fonction de covariance $k(\cdot, \cdot)$. Par conséquent, si n paires de points de données $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ sont observées, nous avons :

$$y_i = f(x_i) + \varepsilon_i \quad (7.20)$$

Où $\{\varepsilon_i\}_{i=1, \dots, n}$ sont des bruits aléatoires normaux indépendants et identiquement distribués d'une moyenne égale à 0 et une variance σ^2 . D'où $(y_1, \dots, y_n)^T$ suit une distribution normale multivariable (n -variable) :

$$(y_1, \dots, y_n)^T \sim \mathcal{N}(\mu, K) \quad (7.21)$$

Où $\mu = (\mu(x_1), \dots, \mu(x_n))^T$ est le vecteur de moyennes et K étant la matrice $n \times n$ de covariance dont le (i, j) -ième élément $K_{ij} = k(x_i, x_j) + \sigma^2 \delta_{ij}$. Ici $\delta_{ij} = 1$ si $i = j$ sinon $\delta_{ij} = 0$.

Supposons un point x^* de test et y^* la valeur de réponse correspondante. Ensuite, la distribution jointe de $(y_1, \dots, y_n, y^*)^T$ est une distribution normale multivariable $((n+1)$ -variable) avec moyenne $\mu = (\mu(x_1), \dots, \mu(x_n), \mu(x^*))^T$ et une matrice de covariance :

$$\begin{bmatrix} K & K^* \\ K^{*T} & k(x^*, x^*) + \sigma^2 \end{bmatrix} \quad (7.22)$$

Ou $K^* = \left(k(x^*, x_1), \dots, k(x^*, x_n) \right)^T$. Donc la distribution conditionnelle de y^* , étant donné que $y = (y_1, \dots, y_n)^T$, est $N(\hat{y}^*, \hat{\sigma}^{*2})$ ou :

$$\hat{y}^* = \mu(x^*) + K^{*T} K^{-1} (y - \mu) ; \quad \hat{\sigma}^{*2} = k(x^*, x^*) + \sigma^2 - K^{*T} K^{-1} K^* \quad (7.23)$$

Une fonction de covariance exponentielle au carré est souvent utilisée dans la littérature :

$$k(x_i, x_j) = \nu \exp \left(-\frac{1}{2} \sum_{d=1}^m w_d (x_{id} - x_{jd})^2 \right) \quad (7.24)$$

Les hyperparamètres $\{\nu, w_1, \dots, w_m\}$ dans l'équation (0.24) et la variance de bruit σ^2 peuvent être estimés par la méthode du maximum de vraisemblance. Le logarithme de la vraisemblance des données d'entraînement est donné par :

$$L(\nu, w_1, \dots, w_m, \sigma^2) = -\frac{1}{2} \log(\det K) - \frac{1}{2} y^T K^{-1} y - \frac{n}{2} \log(2\pi) \quad (7.25)$$

A.1.2 Méthodes de classification

A.1.2.1 Régression logistique (LR)

La régression logistique est utilisée pour une classe de problèmes différente connue sous le nom de problèmes de classification. Cette technique est souvent utilisée lorsque la variable dépendante (cible) est catégorique. Dans le chapitre 6, par exemple, on applique la régression logistique pour prédire s'il y a eu un impact ou non pour chaque test sur piste. Donc, l'objectif consiste essentiellement à prédire le groupe auquel appartient l'objet en cours d'observation. La régression logistique mesure la relation entre la variable dépendante et la ou les variables indépendantes, en estimant ces probabilités. Ensuite, ces probabilités doivent être transformées en valeurs binaires pour pouvoir faire la prédiction.

La matrice des variables indépendantes « caractéristiques » peut être décrite par $X = (x_{ij})_{n \times d}$ où chaque colonne représente une variable indépendante et chaque ligne représente une observation. Étant donné un ensemble de données d'entraînement $\{(X_i, y_i)\}_{i=1}^n$, où $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ et $y_i \in \{0, 1\}$. Si $y_i = 1$ cela signifie que le i-ème échantillon appartient à la classe 1, sinon, il appartient à la classe 2. La transformation logistique du vecteur d'estimation de probabilité $Pr_i = p(y_i = 1 | X_i)$ est modélisée par la fonction logit suivante [54]-[56]:

$$\ln \left[\frac{Pr_i}{1 - Pr_i} \right] = \beta_0 + \sum_{j=1}^d x_{ij}^T \beta_j \quad ; \quad i = 1, 2, \dots, n \quad (7.26)$$

Où β_0 est le biais et β_j est un vecteur de coefficients à estimer. La fonction logarithmique de vraisemblance négative de l'équation (0.26) est donnée par :

$$\ell(\beta) = - \sum_{i=1}^n [y_i \ln(Pr_i) + (1 - y_i) \ln(1 - Pr_i)] \quad (7.27)$$

Minimiser l'équation (7.27) produit l'estimateur du maximum de vraisemblance (MLE) de β . Ensuite, la probabilité de classer le i-ème échantillon est estimé par :

$$Pr_i = \frac{\exp(\beta_0 + \sum_{j=1}^d x_{ij}^T \beta_j)}{1 + \exp(\beta_0 + \sum_{j=1}^d x_{ij}^T \beta_j)} \quad (7.28)$$

L'avantage de la régression logistique est qu'elle offre d'estimer les probabilités Pr_i et $(1 - Pr_i)$ pour chaque classe et de classer chaque résultat simultanément. La règle de classification peut être désignée telle que si : $Pr(y_i = 1 | X_i) \geq 0.5 \Rightarrow y_i$ appartient à la classe 1 et si $Pr(y_i = 1 | X_i) < 0.5 \Rightarrow y_i$ appartient à la classe 2.

A.1.2.2 Classification naïve bayésienne (NB)

La classification naïve bayésienne est une classification basée sur le théorème de Bayes avec une hypothèse d'indépendance entre les variables indépendantes. En d'autres termes, le classifieur naïve bayésien suppose que l'effet de la valeur d'une variable indépendante (x) sur

une classe donnée (y) est indépendant des autres variables indépendantes. Un modèle naïf bayésien est facile à construire, sans aucune estimation de paramètre itérative compliquée, ce qui le rend particulièrement utile pour des problèmes avec un très grand ensemble de données. Outre sa simplicité, le classificateur naïve bayésien est réputé pour surpasser des méthodes de classification sophistiquées.

Le théorème de Bayes offre un moyen de calculer la probabilité a posteriori $P(y|x)$ à partir de $P(y), P(x), P(x|y)$. La règle de Bayes est donnée comme suit :

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \quad (7.29)$$

Ou:

- $P(y|x)$ désigne la probabilité a posteriori de la classe y (cible) sachant x (attribut)
- $P(y)$: Probabilité a priori de la classe y
- $P(x|y)$: Probabilité de vraisemblance de l'attribut x sachant la classe y
- $P(x)$: Probabilité a priori de l'attribut x

À partir de ce modèle de probabilités, une règle de décision doit être ajoutée pour construire un classificateur. L'idée consiste à choisir la plus grande probabilité a posteriori, il s'agit de la règle du maximum a posteriori (MAP) et elle donnée par la fonction suivante [58]-[59]:

$$P(y|x) = \operatorname{argmax}_y P(y) \prod_{i=1}^n P(x_i|y) \quad (7.30)$$

A.1.2.3 Analyse discriminante linéaire et quadratique (LDA, QDA)

L'analyse discriminante linéaire est fréquemment utilisée comme technique de classification ou pour la réduction de la dimensionnalité pour la reconnaissance de formes. Son avantage, par rapport à d'autres techniques de classification telles que les réseaux de neurones et les forêts aléatoires, est la facilité d'interprétation de son modèle. Mathématiquement, les techniques LDA

et QDA modélisent la distribution conditionnelle des données $P(X | y = k)$ pour chaque classe k . Les prédictions peuvent être obtenues en appliquant la règle de Bayes (0.29) [r15]:

$$P(y = k | X) = \frac{P(X | y = k)P(y = k)}{P(X)} = \frac{P(X | y = k)P(y = k)}{\sum_l P(X | y = l)P(y = l)} \quad (7.31)$$

Et on choisit la classe k qui maximise cette probabilité conditionnelle.

Pour l'analyse discriminante linéaire et quadratique, $P(X | y)$ est modélisée comme une distribution gaussienne multivariée avec la densité de probabilité suivante:

$$P(X | y = x) = \frac{1}{(2\pi)^{d/2} |\sum_k|^{1/2}} \exp\left(-\frac{1}{2}(X - \mu_k)^t \sum_k^{-1} (X - \mu_k)\right) \quad (7.32)$$

Où d est le nombre de variables indépendantes (explicatives).

Pour utiliser ce modèle en tant que classificateur, il suffit d'estimer à partir des données d'entraînement [57]:

- Laprobabilité à priori de chaque classe : $P(y = k) = N_k / N$, où N_k est le nombre d'observations appartenant à la classe k et N est le nombre d'observations total.
- La moyenne de chaque classe : $\hat{\mu}_k = \sum_{i=k} x_i / N_k$
- La matrice de covariance : $\sum = \sum_{k=1}^K \sum_{i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T / (N - K)$

Dans le cas de la LDA, on suppose que les gaussiennes de chaque classe partagent la même matrice de covariance: $\sum_k = \sum$ pour tous k . Ce qui conduit à des surfaces de décision linéaires. Les estimations de la QDA sont similaires à celles de la LDA, sauf que des matrices de covariances distinctes doivent être estimées pour chaque classe [57]. Ce qui conduit à des surfaces de décision quadratiques.

A.1.2.4 K-plus proches voisins (K-NN)

La méthode des K-plus proches voisins (K-NN, en anglais : k-nearest neighbours) est une technique de classification non paramétrique qui ne fait aucune hypothèse sur la distribution des

données d'entraînement et ne produit pas explicitement un modèle mathématique. Plutôt, cette technique sauvegarde les données d'entraînement et prédit la classe d'une nouvelle cible en fonction d'une mesure de similarité (par exemple, fonction de distance).

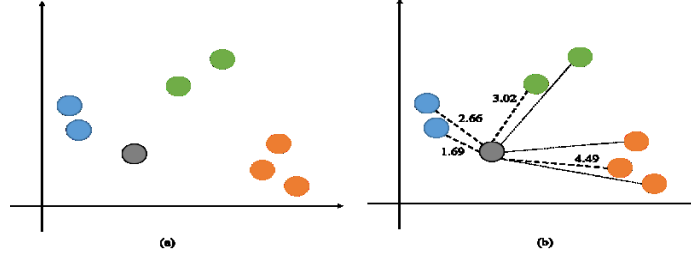


Figure 7-4 Exemple de classification [K-NN]

Pour classer un objet non étiqueté, la distance de cet objet par rapport aux objets étiquetés est calculée, ses k -voisins les plus proches sont identifiés et l'étiquette de classe de la majorité des voisins les plus proches est ensuite utilisée pour déterminer la classe (étiquette) de l'objet. Prenons l'exemple de la Figure 7-4-(a), nous voulons classer le point gris dans l'une des trois classes: bleu, vert ou orange. Nous commençons par calculer les distances entre le point gris et le k -plus proches voisins ($k = 4$ dans cet exemple) (voir Figure 7-4-(b)). Pour les variables d'entrée à valeurs réelles, la mesure de distance la plus populaire est la distance Euclidienne [60]-[62]. Les autres mesures de distance populaires sont: Manhattan et Minkowski. En cas de variables catégoriques, c'est la distance de Hamming qui est souvent utilisée.

$$\text{Euclidienne : } D_E = \sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

$$\text{Manhattan : } D_{Mh} = \sum_{i=1}^k |x_i - y_i|$$





$$\text{Minkowski : } D_{Mk} = \left(\sum_{i=1}^k (|x_i - y_i|)^q \right)^{1/q}$$

$$\text{Hamming : } D_H = \sum_{i=1}^k |x_i - y_i| \quad (7.33)$$

$$\text{Si : } x = y \Rightarrow D = 0$$

$$\text{Si : } x \neq y \Rightarrow D = 1$$

Table 7-1 Exemple de classification [K-NN]

Point	Distance	Classe	# de votes
	1.69	Bleu	2
	2.66	Vert	1
	3.02	Orange	1
	4.49		

La classe bleue gagne le vote, ce qui fait que le point gris est classé comme point bleu. Trouver la valeur de k n'est pas une tâche facile. Cela dépend beaucoup des caractéristiques du problème à résoudre. Parfois, il est préférable de passer en revue plusieurs valeurs possibles pour k pour en choisir la plus optimale.

Annexe B

Documents pertinents actuels et au niveau mondial relatif à la normalisation des véhicules intelligents/connectés/autonomes

B.1 National Highway Traffic Safety Administration (NHTSA)

B.1.1 Safety Technologies

- Automatic Emergency Braking Systems (AEB) [88]
- Dynamic Brake Support (DBS) versus Crash Imminent Braking (CIB) [89]
- Rearview Video System [90]
- Forward Collision Warning System (FCW) [91]-[92]
- Lane Departure Warning System (LDW) and Lane Keeping Support (LKS) [93]-[95]
- Blind Spot Warning/Monitoring [96]

B.1.2 Vehicle to Vehicle Communications

- V2V Communications : FMVSS No. 150 [97]
- Commercial Connected Vehicle Test Procedure Development and Test Results [95]
- Characterization Test Procedures for Intersection Collision Avoidance Systems Based on Vehicle-to-Vehicle Communications [98]

B.1.3 Vehicle Cybersecurity

- NHTSA's Approach and Guidance Release [99]-[100]

B.1.4 Automated Vehicles

- Federal Automated Vehicles Policy Overview [25]
- Preparing for the Future of Transportation: Automated Vehicles 3.0 [156]

B.2 European New Car Assessment Program (Euro NCAP)

B.2.1 Safety Assist

- Euro NCAP Assessment Protocol - Safety Assist - v8.0.2 [101]
- Euro NCAP Test Protocol - AEB Systems - v2.0.1 [102]
- Euro NCAP Test Protocol - Lane Support Systems - v.2.1.0 [103]
- Euro NCAP Test Protocol - Speed Assist Systems - v2.0 [104]

B.2.2 Vulnerable Road User (VRU) Protection

- Euro NCAP Assessment Protocol - Pedestrian Protection – v9.0.2 [105]
- Euro NCAP Pedestrian Testing Protocol – v8.4 [106]
- Euro NCAP Test Protocol - AEB VRU Systems - v.2.0.2 [107]

B.3 International Organization For Standardization (ISO)

B.3.1 Advanced Driver Assistance Systems

- ISO 15623:2013 / Forward vehicle collision warning systems -- Performance requirements and test procedures [108]
- ISO 15622:2018 / Adaptive cruise control systems [109]
- ISO 17361:2017 / Lane departure warning [110]
- ISO 11270:2014 / Lane keeping assistance systems (LKAS) [111]
- ISO 17387:2008 / Lane change decision aid systems (LCDAS) [112]
- ISO 16787:2017 / Assisted parking system (APS) [113]

B.3.2 Vehicle To X Communications

- ISO/TS 15624:2001 / Traffic Impediment Warning Systems (TIWS) -- System requirements [114]
- ISO/TS 19091:2017 / Cooperative ITS -- Using V2I and I2V communications for applications related to signalized intersections [115]
- ISO 22951:2009 / Data dictionary and message sets for preemption and prioritization signal systems for emergency and public transport vehicles (PRESTO) [116]
- ISO 26684:2015 / Cooperative intersection signal information and violation warning systems (CIWS) -- Performance requirements and test procedures [117]
- ISO/TS 20026:2017 / Cooperative ITS -- Test architecture [118]
- ISO/TS 19321:2015/ Dictionary of in-vehicle information (IVI) data structures [119]
- ISO 18750:2018 / Cooperative ITS -- Local dynamic map [120]
- ISO 17419:2018 / Cooperative systems -- Globally unique identification [121]
- ISO 17423:2018 / Cooperative systems -- Application requirements and objectives [122]
- ISO/TS 17425:2016 / Cooperative systems -- Data exchange specification for in-vehicle presentation of external road and traffic related data [123]
- ISO/TS 17426:2016 / Cooperative systems -- Contextual speeds [124]
- ISO 17427-1:2018 / Cooperative ITS -- Part 1: Roles and responsibilities in the context of co-operative ITS architecture(s) [125]

B.3.3 Automated Vehicles

- ISO/TR 20545:2017 / Vehicle/roadway warning and control systems -- Report on standardisation for vehicle automated driving systems (RoVAS)/Beyond driver assistance systems [126]

B.4 Society of Automotive Engineers (SAE)

B.4.1 Active Safety Systems Standards Committee

- SAE J3087 / Automatic Emergency Braking (AEB) test methods and performance assessment [127]
- SAE J3088 / Active Safety System Sensors [128]
- SAE J3116 / Active Safety Pedestrian Test Mannequin Recommendation [129]
- SAE J3063 / Active Safety Systems Terms & Definitions [130]

B.4.2 Safety and Human Factors standards Steering Committee

- SAE J2808 / Lane Departure Warning Systems: Information for the Human [131]
- SAE J3048 / Driver-Vehicle Interface Considerations for Lane Keeping Assistance Systems [132]
- SAE J2396 / Definitions and Experimental Measures Related to the Specification of Driver Visual Behaviour Using Video Based Techniques [133]
- SAE J2365 / Calculation and Measurement of the Time to Complete In-Vehicle Navigation and Route Guidance Tasks [134]
- SAE J2802 / Blind Spot Monitoring System (BSMS): Operating Characteristics and User Interface [135]
- SAE J3114 / Human Factors Definitions for Automated Driving and Related Research Topics (Ground Vehicle Standard) [136]
- SAE J2830 / Process for Comprehension Testing of In-Vehicle Symbols [137]
- SAE J3077 / Definitions and Data Sources for the Driver Vehicle Interface [138]
- SAE J2889 / Measurement of Minimum Noise Emitted by Road Vehicles [139]
- SAE J2395 / In-Vehicle Message Priority (WIP = Work In Progress) [140]
- SAE J2831 / Development of Design and Engineering Recommendations for In-Vehicle Alphanumeric Messages (WIP = Work In Progress) [141]
- SAE J2944 / Operational Definitions of Driving Performance Measures and Statistics [142]

- SAE J2364 / Navigation and Route Guidance Function Accessibility While Driving [143]
- SAE J2988 / Guidelines for Speech Input and Audible Output in a Driver Vehicle Interface [144]
- SAE J2400 / Human Factors in Forward Collision Warning Systems: Operating Characteristics and User Interface Requirements (WPI = Work In Progress) [145]
- SAE J2399 / Adaptive Cruise Control (ACC) Operating Characteristics and User Interface [146]
- SAE J2972_201403 / Definition of Road Vehicle Hands-Free Operation of a Person-to-Person Wireless Communication System or Device [147]

B.4.3 DSRC Tech Committee (Dedicated Short Range Communication)

- SAE J2945/9 / Vulnerable Road User Safety Message Minimum Performance Requirements [148]
- SAE J2735ASN / Dedicated Short Range Communications (DSRC) Message Set Dictionary™ ASN file [149]
- SAE J2735SET / Dedicated Short Range Communications (DSRC) Message Set Dictionary™ Set [150]
- SAE J2945/1 / On-Board System Requirements for V2V Safety Communications [151]
- SAE J2945/2 / Dedicated Short Range Communications (DSRC) Performance Requirements for V2V Safety Awareness (Ground Vehicle Standard) [152]
- SAE J3067 / Candidate Improvements to Dedicated Short Range Communications (DSRC) Message Set Dictionary [SAE J2735] Using Systems Engineering Methods [153]

B.4.4 On-Road Automated Driving (ORAD) Committee

- SAE J3016 / Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles [41]

- SAE J3092 / Dynamic Test Procedures for Verification & Validation of Automated Driving Systems (ADS) (WIP = Work In Progress) [154]
- SAE J3018_201503 / Guidelines for Safe On-Road Testing of SAE Level 3, 4, and 5 Prototype Automated Driving Systems (ADS) [155]

Annexe C

Proposition d'une plateforme de test et validation MIL/SIL/PIL/HIL pour PMG Technologies

C.1 Objectifs de la plateforme

- Réduire le temps des tests sur piste/route
- Accélération du prototypage d'algorithmes en générant le code C et C++
- Vérification des algorithmes déployés en automatisant les tests HIL en temps réel

C.2 Principales caractéristiques de la plateforme

- Développer, gérer et exécuter des tests basés sur la simulation
- Construire, exécuter et tester des applications en temps réel
- Effectuer des tests en boucle fermée d'algorithmes ADAS en simulation
- Déboguer les problèmes matériels en comparant les données en temps réel avec les données de simulation
- Régler et calibrer les algorithmes ADAS
- Automatiser et générer les rapports des tests HIL
- Générer le code C et C++ à partir des modèles Simulink, Stateflow et fonctions Matlab

C.3 Logiciels nécessaires

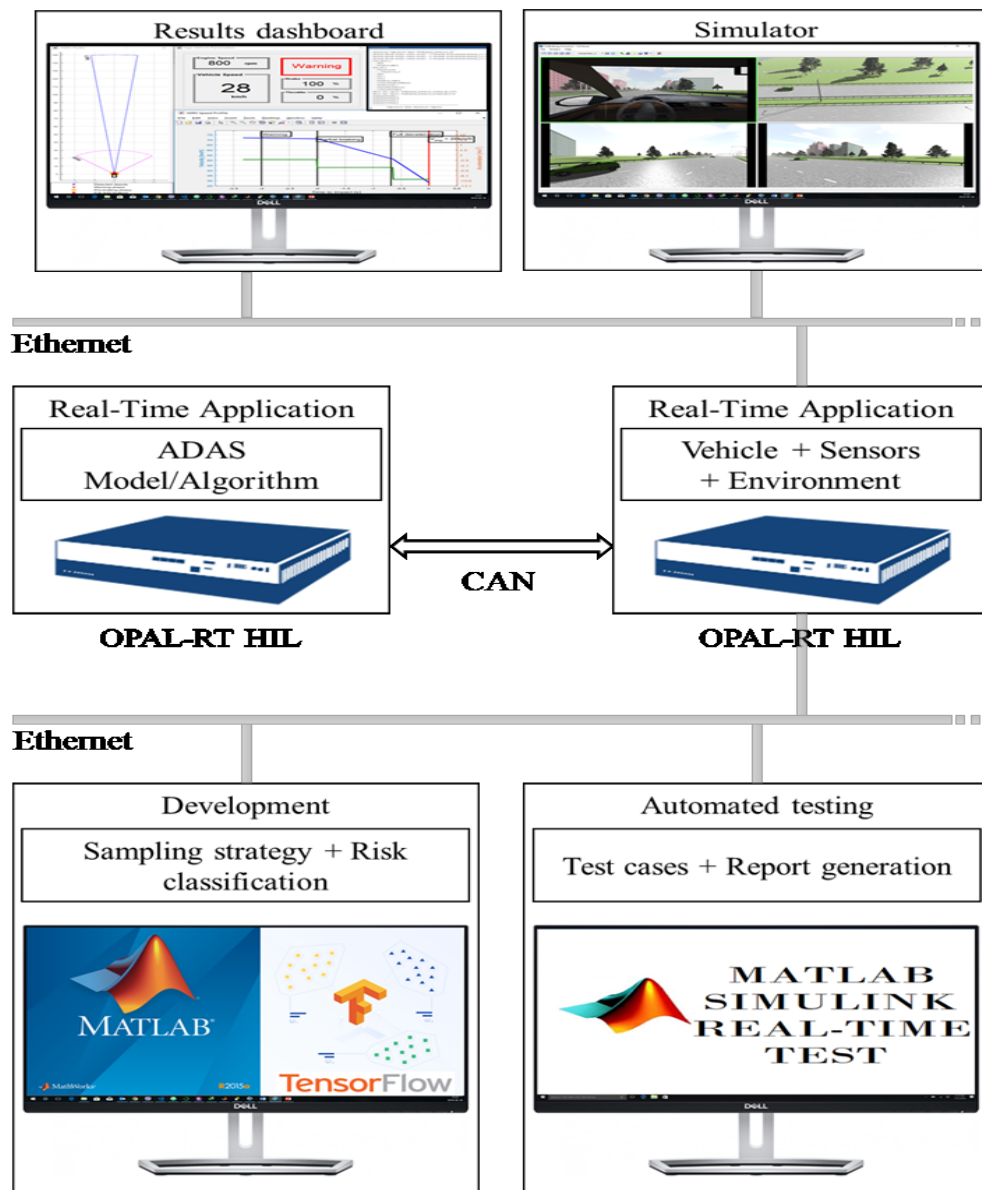
- Matlab Simulink, Simulink Real-Time, Simulink Test, Simulink Coder, Simscape.
- Simulateur de conduite : PreScan, Pro-SiVIC, etc. (L'idéal serait d'avoir plusieurs simulateurs à disposition).

C.4 Matériels nécessaires

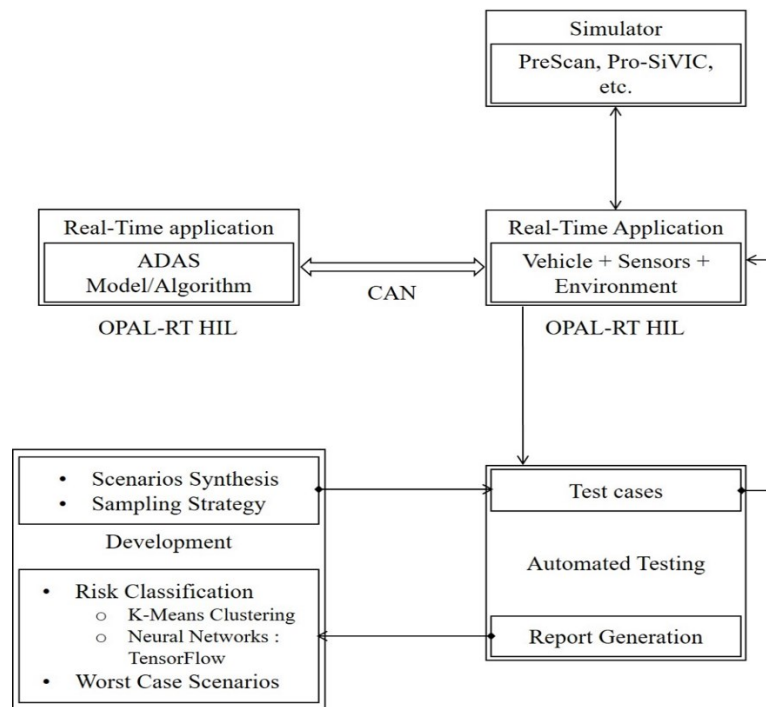
- 2 stations : 1 pour les simulateurs + 1 pour le développement et l'automatisation des tests

- 2 plateformes OPAL-RT HIL
- 2 cartes d'interfaçages + 2 connecteurs CAN-Bus
- 1 hub Ethernet

C.5 Représentation visuelle de la plateforme



C.6 Schéma de fonctionnement la plateforme



C.7 Clients potentiels

- Communauté académique : Universités, laboratoires de recherche, etc.
- Entreprises : R&D, innovation, etc.

C.8 Estimation budgétaire

	Prix (USD)	License/annuelle (USD)	Estimation (USD)
Hardware HIL	*	*	*
1 simulateur de conduite		*	*
2 stations	*	*	*
Matlab, Matlab Simulink, Simulink Real-Time, Simulink Test, Simulink Coder, imscape		*	*
Estimation Totale (+15% marge)			*

Références bibliographiques

- [1] M. Wetterwald, “Normes européennes dans les STI coopératifs,” *Ref: TIP601WEB - “Véhicule et mobilité du futur,”* 10-Feb-2015. [Online]. Available: <https://www.techniques-ingenieur.fr/base-documentaire/ingenierie-des-transportsth14/systemes-de-transport-intelligents-42575210/normes-europeennes-dans-les-sti-cooperatifs-trp1007/>. [Accessed: 19-Jan-2019].
- [2] L. Greer, J. Fraser, D. Hicks, M. Mercer, and K. Thompson, “Intelligent Transportation Systems Benefits, Costs, and Lessons Learned: 2018 Update Report,” U.S Department of Transportation, Technical Report FHWA-JPO-18-641, Mar. 2018.
- [3] M. Bergeron *et al.*, *Vocabulaire trilingue des systèmes de transport intelligents = trilingual vocabulary of intelligent transportation systems = vocabulario trilingüe de los sistemas inteligentes de transporte*. Montréal: Office québécois de la langue française, 2009.
- [4] W. Wachenfeld and H. Winner, “The Release of Autonomous Vehicles,” in *Autonomous Driving*, M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 425–449.
- [5] L. Fraade-Blanar, *Measuring automated vehicle safety: forging a framework*. Santa Monica, CA: RAND Corporation, 2018.
- [6] E. Bringmann and A. Krämer, “Model-Based Testing of Automotive Systems,” in *2008 1st International Conference on Software Testing, Verification and Validation*, 2008, pp. 485–493.
- [7] P. Skruch and G. Buchala, “Model-Based Real-Time Testing of Embedded Automotive Systems,” *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, vol. 7, no. 2, pp. 337–344, Apr. 2014.
- [8] S. Riedmaier, J. Nesensohn, C. Gutenkunst, T. Duser, B. Schick, and H. Abdellatif, “Validation of X-in-the-Loop Approaches for Virtual Homologation of Automated Driving Functions,” *AVL - Development, testing & simulation of powertrain systems*, p. 12, Sep-2018.

- [9] D. J. Verburg, A. C. M. van der Knaap, and J. Ploeg, "VEHIL: developing and testing intelligent vehicles," in *IEEE Intelligent Vehicle Symposium, 2002*, 2002, vol. 2, pp. 537–544 vol.2.
- [10] O. J. Gietelink, J. Ploeg, B. D. Schutter, and M. Verhaegen, "VEHIL: Test facility for fault management testing of advanced driver assistance systems," presented at the ITS World Congress, Madrid, Spain, 2003, p. 14.
- [11] O. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, "VEHIL: A Test Facility for Validation of Fault Management Systems for Advanced Driver Assistance Systems," *IFAC Proceedings Volumes*, vol. 37, no. 22, pp. 397–402, Apr. 2004.
- [12] P. Skruch, R. Dlugosz, K. Kogut, P. Markiewicz, D. Sasin, and M. Rózewicz, "The Simulation Strategy and Its Realization in the Development Process of Active Safety and Advanced Driver Assistance Systems," presented at the SAE 2015 World Congress & Exhibition, 2015.
- [13] L. Raffaelli, F. Vallée, G. Fayolle, P. D. Souza, X. Rouah, and et al, "Facing ADAS validation complexity with usage oriented testing," in *8th European Congress - ERTS 2016*, Toulouse, France, 2016.
- [14] L. Raffaelli *et al.*, "Facing ADAS validation complexity with usage oriented testing," *CoRR*, vol. abs/1607.07849, 2016.
- [15] J. Jansson, *Collision avoidance theory: with application to automotive collision mitigation*. Linköping: Univ, 2005.
- [16] O. J. Gietelink and I. and L. Netherlands Research School for Transport, "Design and validation of advanced driver assistance systems," Netherlands TRAIL Research School, Delft, The Netherlands, 2007.
- [17] D. Zhao, "Accelerated Evaluation of Automated Vehicles," Mechanical Engineering, University of Michigan, 2016.
- [18] "Home," *PMG Technologies*. [Online]. Available: <https://www.pmgtest.com/>. [Accessed: 06-Nov-2018].
- [19] D. Bezzina and J. Sayer, "Safety pilot model deployment: Test conductor team report," Washington, DC: National Highway Traffic Safety Administration, DOT HS 812 171, Jun. 2015.

- [20] J. Sayer *et al.*, “Integrated Vehicle-Based Safety Systems: Field Operational Test Final Program Report,” NHTSA, DOT HS, DOT HS 811 482, 2011.
- [21] D. LeBlanc *et al.*, “Road Departure Crash Warning System Field Operational Test: Methodology and Results,” Washington, DC: National Highway Traffic Safety Administration, UMTRI-2006-9-1, Jun. 2006.
- [22] J. Orban, J. Hadden, G. Stark, and V. Brown, “Evaluation of the Mack Intelligent Vehicle Initiative Field Operational Test : Final Report,” Federal Motor Carrier Safety Administration, FMCSA-06-016, 2006.
- [23] J. M. Menéndez, J. Alfonso, E. Cacheiro, and N. Sánchez, “Cooperative ITS communications architecture: the FOTsis project approach and beyond,” *IET Intelligent Transport Systems*, vol. 9, no. 6, pp. 591–598, Aug. 2015.
- [24] P. Koopman and M. Wagner, “Challenges in Autonomous Vehicle Testing and Validation,” *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, Apr. 2016.
- [25] “Federal Automated Vehicles Policy - September 2016,” *US Department of Transportation*, 19-Sep-2016. [Online]. Available: <https://www.transportation.gov/AV/federal-automated-vehicles-policy-september-2016>. [Accessed: 06-Nov-2018].
- [26] N. E. Chelbi, D. Gingras, and C. Sauvageau, “Modified Metropolis-Hastings Algorithm for Efficient Sampling from Discrete Probability Distributions (MMH-DPD) Applied to Field Operational Tests database (SPMD),” in *IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, 2018, pp. 125–131.
- [27] N. E. Chelbi, D. Gingras, and C. Sauvageau, “New Field Operational Tests Sampling Strategy Based on Metropolis-Hastings Algorithm,” in *Intelligent Systems and Applications. IntelliSys 2018. Advances in Intelligent Systems and Computing*, 2019, vol. 868.
- [28] N. E. Chelbi, D. Gingras, and C. Sauvageau, “Proposal of a new virtual evaluation approach of preventive safety applications and advanced driver assistance functions – application: AEB system,” *IET Intelligent Transport Systems*, vol. 12, no. 9, pp. 1148–1156, 2018.

- [29] N. E. Chelbi, D. Gingras, and C. Sauvageau, “Worst-case Scenarios Identification Approach for the Evaluation of Advanced Driver Assistance Systems in Intelligent/Autonomous Vehicles under Multiple Conditions,” *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, p. 34.
- [30] “Véhicules automobiles intelligents,” *Archives*. [Online]. Available: <http://archives.cnrs.fr/presse/article/411>. [Accessed: 21-Jan-2019].
- [31] D. Gingras and A. de La Fortelle, “Large-scale deployment of autonomous vehicles: still a long road ahead,” *Unpublished*.
- [32] A. Amditis, P. Lytrivis, and E. Portouli, “Sensing and Actuation in Intelligent Vehicles,” in *Handbook of Intelligent Vehicles*, A. Eskandarian, Ed. London: Springer London, 2012, pp. 31–60.
- [33] A. Broggi, P. Grisleri, and P. Zani, “Sensors technologies for intelligent vehicles perception systems: A comparison between vision and 3D-LIDAR,” in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, Netherlands, 2013, pp. 887–892.
- [34] “ADAS :: La sécurité routière de A à Z.” [Online]. Available: <https://www.securite-routiere-az.fr/a/adas/>. [Accessed: 21-Jan-2019].
- [35] D. Gingras, “Les assureurs face à la conduite automatisée et au déploiement des véhicules autonomes,” presented at the Groupement des assureurs automobiles, Montréal, Oct-2016.
- [36] “Automated Driving Roadmap,” European Road Transport Research Advisory Council (ERTRAC), Jul. 2015.
- [37] “Roadmap on Smart Systems For Automated Driving,” European Technology Platform on Smart Systems Integration (EPoSS), Berlin, Apr. 2015.
- [38] M. Oonk and J. Svensson, “Roadmap : Automation in Road Transport,” ERTICO ITS Europe and Volvo Group Truck Technology, Version 1.0, May 2013.
- [39] S. Thrun *et al.*, “Stanley: The robot that won the DARPA Grand Challenge,” *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, Sep. 2006.
- [40] F. Bounini, “Simulateur Temps-Réel De Véhicules Intelligents Et Autonomes En Conduite Collaborative,” Université de Sherbrooke, Canada, 2018.

- [41] “Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems (J3016 Ground Vehicle Standard) - SAE.” [Online]. Available: https://saemobilus.sae.org/content/j3016_201401. [Accessed: 21-Jan-2019].
- [42] S. Cherkaoui, “Étude spécialisée GEI-705, Communication Véhiculaire,” presented at the Département de génie électrique et génie informatique, Université de Sherbrooke., 2016.
- [43] G. James, D. Witten, T. Hastie, and R. Tibshirani, Eds., *An introduction to statistical learning: with applications in R*. New York: Springer, 2013.
- [44] S. Jia, C. Hou, and J. Wang, “Software aging analysis and prediction in a web server based on multiple linear regression algorithm,” in *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, Guangzhou, 2017, pp. 1452–1456.
- [45] S. IIT Kanpur, “Chapter 3 : Multiple Linear Regression Model,” in *Econometrics*, p. 41.
- [46] J. Hawkins, “Chapter 4: Support Vector Regression,” in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, pp. 67–80.
- [47] H. Nourali and M. Osanloo, “Mining capital cost estimation using Support Vector Regression (SVR),” *Resources Policy*, Nov. 2018.
- [48] A. García-Floriano, C. López-Martín, C. Yáñez-Márquez, and A. Abran, “Support vector regression for predicting software enhancement effort,” *Information and Software Technology*, vol. 97, pp. 99–109, May 2018.
- [49] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, Apr. 2011.
- [50] R. Wu and B. Wang, “Gaussian process regression method for forecasting of mortality rates,” *Neurocomputing*, vol. 316, pp. 232–239, Nov. 2018.
- [51] S. Das, S. Roy, and R. Sambasivan, “Fast Gaussian Process Regression for Big Data,” *Big Data Research*, vol. 14, pp. 12–26, Dec. 2018.
- [52] P. L. T. Duong, X. Xu, Q. Yang, and N. Raghavan, “Gaussian process regression approach for robust design and yield enhancement of self-assembled nanostructures,” *Microelectronics Reliability*, vol. 88–90, pp. 85–90, Sep. 2018.

- [53] B. Wang, T. Chen, and A. Xu, “Gaussian process regression with functional covariates and multivariate response,” *Chemometrics and Intelligent Laboratory Systems*, vol. 163, pp. 1–6, Apr. 2017.
- [54] H.-L. Hsu, Y. I. Chang, and R.-B. Chen, “Greedy active learning algorithm for logistic regression models,” *Computational Statistics & Data Analysis*, vol. 129, pp. 119–134, Jan. 2019.
- [55] O. S. Qasim and Z. Y. Algamal, “Feature selection using particle swarm optimization-based logistic regression model,” *Chemometrics and Intelligent Laboratory Systems*, vol. 182, pp. 41–46, Nov. 2018.
- [56] Z. Y. Algamal and M. H. Lee, “Penalized logistic regression with the adaptive LASSO for gene selection in high-dimensional cancer classification,” *Expert Systems with Applications*, vol. 42, no. 23, pp. 9326–9332, Dec. 2015.
- [57] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning Data Mining, Inference, and Prediction* Valerie and Patrick Hastie, Second Edition. Springer, 2017.
- [58] N. Sun, B. Sun, J. (Denny) Lin, and M. Y.-C. Wu, “Lossless Pruned Naive Bayes for Big Data Classifications,” *Big Data Research*, vol. 14, pp. 27–36, Dec. 2018.
- [59] O. Tejaswini, “New Feature Selection Process to Enhance Naïve Bayes Classification,” *IEEE Conference Record*, p. 4, 2018.
- [60] Y. Zhang, G. Cao, B. Wang, and X. Li, “A novel ensemble method for k-nearest neighbor,” *Pattern Recognition*, vol. 85, pp. 13–25, Jan. 2019.
- [61] J. Gou, H. Ma, W. Ou, S. Zeng, Y. Rao, and H. Yang, “A generalized mean distance-based k-nearest neighbor classifier,” *Expert Systems with Applications*, vol. 115, pp. 356–372, Jan. 2019.
- [62] Ö. F. Ertuğrul and M. E. Tağluk, “A novel version of k nearest neighbor: Dependent nearest neighbor,” *Applied Soft Computing*, vol. 55, pp. 480–490, Jun. 2017.
- [63] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, 1st ed. Chapman and Hall/CRC, 2012.
- [64] S. Raschka and V. Mirjalili, *Python machine learning: machine learning and deep learning with Python, scikit-learn, and TensorFlow*, Second edition. Birmingham Mumbai: Packt Publishing Ltd, 2017.

- [65] “PreScan,” *TASS International*, 31-Dec-2012. [Online]. Available: <https://tass.plm.automation.siemens.com/prescan>. [Accessed: 22-Jan-2019].
- [66] “Civitec - Sensor Simulation Solutions.” [Online]. Available: <http://www.civitec.com/>. [Accessed: 22-Jan-2019].
- [67] “Real-Time simulation | Real-Time Solutions | OPAL-RT,” *OPAL-RT*. [Online]. Available: <https://www.opal-rt.com/>. [Accessed: 22-Jan-2019].
- [68] “Industries & Solutions - dSPACE.” [Online]. Available: <https://www.dspace.com/en/inc/home/applicationfields/stories.cfm#filterterms=term-133>. [Accessed: 22-Jan-2019].
- [69] “Mechanical Simulation.” [Online]. Available: <https://www.carsim.com/>. [Accessed: 22-Jan-2019].
- [70] “Products,” *AVSimulation*. .
- [71] H.-H. Yang and H. Peng, “Development of an errorable car-following driver model,” *Vehicle System Dynamics*, vol. 48, no. 6, pp. 751–773, Jun. 2010.
- [72] J. Woodrooffe *et al.*, “Performance Characterization and Safety Effectiveness Estimates of Forward Collision Avoidance and Mitigation Systems for Medium/Heavy Commercial Vehicles - Final Report -,” The University of Michigan Transportation Research Institute, UMTRI-2011-36, Aug. 2012.
- [73] R. Karlsson, J. Jansson, and F. Gustafsson, “Model-based statistical tracking and decision making for collision avoidance application,” in *Proceedings of the 2004 American Control Conference*, Boston, MA, USA, 2004, pp. 3435–3440 vol.4.
- [74] O. J. Gietelink, B. D. Schutter, and M. Verhaegen, “A probabilistic approach for validation of advanced driver assistance systems,” presented at the Proceedings of the 84th Annual Meeting of the Transportation Research Board, Washington, DC, 2005, p. 24.
- [75] A. B. Owen, “Chapter 9: Importance sampling,” in *Monte Carlo theory, methods and examples*, 2013, p. 46.
- [76] R. Srinivasan, *Importance Sampling: Applications in Communications and Detection*. Berlin Heidelberg: Springer-Verlag, 2002.

- [77] D. Zhao *et al.*, “Accelerated Evaluation of Automated Vehicles Safety in Lane-Change Scenarios Based on Importance Sampling Techniques,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 595–607, Mar. 2017.
- [78] X. Zhu, “Markov Chain Monte Carlo,” in *Advanced Artificial Intelligence (“Machine Learning 2”)*, 2011.
- [79] A. Fischer, “An Uncoupling-Coupling Method for Markov Chain Monte Carlo Simulations with an Application to Biomolecules,” phd, Free University of Berlin, 2003.
- [80] S. Chib, “Chapter 57 - Markov Chain Monte Carlo Methods: Computation and Inference,” in *Handbook of Econometrics*, vol. 5, J. J. Heckman and E. Leamer, Eds. Elsevier, 2001, pp. 3569–3649.
- [81] T. C. Gouvernement du Canada, “Règlement sur la sécurité des véhicules automobiles (C.R.C., ch. 1038),” 09-Nov-2009. [Online]. Available: <https://www.tc.gc.ca/fra/lois-reglements/reglements-crc-ch1038.htm>. [Accessed: 22-Jan-2019].
- [82] “Federal Motor Vehicle Safety Standards and Regulations.” [Online]. Available: <https://icsw.nhtsa.gov/cars/rules/import/FMVSS/>. [Accessed: 22-Jan-2019].
- [83] “Search vehicle safety ratings,” *NHTSA*, 10-Jun-2016. [Online]. Available: <https://www.nhtsa.gov/ratings>. [Accessed: 22-Jan-2019].
- [84] “The European New Car Assessment Programme | Euro NCAP.” [Online]. Available: <https://www.euroncap.com:443/en>. [Accessed: 22-Jan-2019].
- [85] “Euro NCAP,” *Wikipédia*. 10-Jan-2019.
- [86] “Organisation internationale de normalisation,” *Wikipédia*. 14-Nov-2018.
- [87] “SAE International.” [Online]. Available: <https://www.sae.org/>. [Accessed: 22-Jan-2019].
- [88] G. J. Forkenbrock and A. S. Synder, “NHTSA’s 2014 automatic emergency braking test track evaluations,” Washington, DC: National Highway Traffic Safety Administration, Report No. DOT HS 812 166.
- [89] “Dynamic Brake Support Performance Evaluation (Working Draft),” U. S. Department of Transportation, Aug. 2014.
- [90] “New Car Assessment Program - Rearview Video System Confirmation Test -,” Office of Crash Avoidance Standards (NVS-120) National Highway Traffic Safety Administration, Sep. 2013.

- [91] “Forward Collision Warning System Confirmation Test,” Office of Vehicle Safety Office of Crash Avoidance Standards National Highway Traffic Safety Administration, Feb. 2013.
- [92] G. Howe, G. Xu, R. Hoover, D. Elsasser, and F. Barichman, “Commercial connectedvehicle test procedure development and test results – Forward collision warning,” Washington, DC: National Highway Traffic Safety Administration, Report No. DOT HS 812 298, Jul. 2016.
- [93] “Lane Departure Warning System Confirmation Test And Lane Keeping Support Performance Documentation,” Office of Vehicle Safety Office of Crash Avoidance Standards, Feb. 2013.
- [94] R. L. Hoover, J. R. Sughosh, G. Howe, and F. S. Barickman, “Heavy-vehicle lane departure warning test development,” Washington, DC: National Highway Traffic Safety Administration, Report No. DOT HS 812078, Nov. 2014.
- [95] G. Howe, G. Xu, and D. Elsasser, “Commercial Connected Vehicle Test Procedure Development and Test Results – Blind Spot Warning/Lane Change Warning,” Washington, DC: National Highway Traffic Safety Administration, Report No. DOT HS 812 317, Nov. 2016.
- [96] G. Forkenbrock, T. R. V. Buskirk, and M. Heitz, “Blind Spot Monitoring in Light Vehicles - System Performance,” Washington, DC: National Highway Traffic Safety Administration, Report No. DOT HS 812 045, Jul. 2014.
- [97] “Federal Motor Vehicle Safety Standards; V2V Communications,” DOT National Highway Traffic Safety Administration, Docket No. NHTSA–2016–0126, Jan. 2017.
- [98] T. Tiernan, S. Toma, W. Najm, and O. Altan, “Characterization Test Procedures for Intersection Collision Avoidance Systems Based on Vehicle-to-Vehicle Communications,” Washington, DC: National Highway Traffic Safety Administration, Report No. DOT HS 812 223, Dec. 2015.
- [99] “Cybersecurity best practices for modern vehicles,” Washington, DC: National Highway Traffic Safety Administration, Report No. DOT HS 812 333, Oct. 2016.
- [100] “NHTSA and Vehicle Cybersecurity,” NHTSA, 12707b-101916-v5c, 2016.
- [101] “Assessment Protocol - Safety Assist - Version 8.0.2,” European New Car Assessment Programme (Euro Ncap), Nov. 2017.

- [102] “TEST PROTOCOL - AEB systems - Version 2.0.1,” European New Car Assessment Programme (Euro Ncap), Nov. 2017.
- [103] “TEST PROTOCOL – Lane Support Systems - Version 2.0.1,” European New Car Assessment Programme (Euro Ncap), Nov. 2017.
- [104] “TEST PROTOCOL – SPEED ASSIST SYSTEMS - Version 2.0,” European New Car Assessment Programme (Euro Ncap), Nov. 2017.
- [105] “ASSESSMENT PROTOCOL – PEDESTRIAN PROTECTION - Version 9.0.2,” European New Car Assessment Programme (Euro Ncap), Nov. 2017.
- [106] “PEDESTRIAN TESTING PROTOCOL - Version 8.4,” European New Car Assessment Programme (Euro Ncap), Nov. 2017.
- [107] “TEST PROTOCOL – AEB VRU systems - Version 2.0.2,” European New Car Assessment Programme (Euro Ncap), Nov. 2017.
- [108] “ISO 15623:2013,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/05/66/56655.html>. [Accessed: 23-Jan-2019].
- [109] “ISO 15622:2018,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/15/71515.html>. [Accessed: 23-Jan-2019].
- [110] “ISO 17361:2017,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/23/72349.html>. [Accessed: 23-Jan-2019].
- [111] “ISO 11270:2014,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/05/03/50347.html>. [Accessed: 23-Jan-2019].
- [112] “ISO 17387:2008,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/04/36/43654.html>. [Accessed: 23-Jan-2019].
- [113] “ISO 16787:2017,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/37/73768.html>. [Accessed: 23-Jan-2019].

- [114] “ISO/TS 15624:2001,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/02/78/27833.html>. [Accessed: 23-Jan-2019].
- [115] “ISO/TS 19091:2017,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/98/69897.html>. [Accessed: 23-Jan-2019].
- [116] “ISO 22951:2009,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/04/60/46009.html>. [Accessed: 23-Jan-2019].
- [117] “ISO 26684:2015,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/04/36/43653.html>. [Accessed: 23-Jan-2019].
- [118] “ISO/TS 20026:2017,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/33/73360.html>. [Accessed: 23-Jan-2019].
- [119] “ISO/TS 19321:2015,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/46/64606.html>. [Accessed: 23-Jan-2019].
- [120] “ISO 18750:2018,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/94/69433.html>. [Accessed: 23-Jan-2019].
- [121] “ISO 17419:2018,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/00/70077.html>. [Accessed: 23-Jan-2019].
- [122] “ISO 17423:2018,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/82/68267.html>. [Accessed: 23-Jan-2019].
- [123] “ISO/TS 17425:2016,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/05/97/59723.html>. [Accessed: 23-Jan-2019].

- [124] “ISO/TS 17426:2016,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/05/97/59724.html>. [Accessed: 23-Jan-2019].
- [125] “ISO 17427-1:2018,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/69/66924.html>. [Accessed: 23-Jan-2019].
- [126] “ISO/TR 20545:2017,” *ISO*. [Online]. Available: <http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/83/68300.html>. [Accessed: 23-Jan-2019].
- [127] “Automatic Emergency Braking (AEB) System Performance Testing (J3087 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j3087_201710. [Accessed: 23-Jan-2019].
- [128] “Active Safety System Sensors (J3088 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j3088_201711. [Accessed: 23-Jan-2019].
- [129] “Active Safety Pedestrian Test Mannequin Recommendation (J3116 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j3116_201706. [Accessed: 23-Jan-2019].
- [130] “Active Safety Systems Terms & Definitions (J3063 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j3063_201511. [Accessed: 23-Jan-2019].
- [131] “Lane Departure Warning Systems: Information for the Human Interface (J2808 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2808_201701. [Accessed: 23-Jan-2019].
- [132] “Driver-Vehicle Interface Considerations for Lane Keeping Assistance Systems (J3048 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/J3048_201602/. [Accessed: 23-Jan-2019].
- [133] “Definitions and Experimental Measures Related to the Specification of Driver Visual Behaviour Using Video Based Techniques (J2396 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/J2396_201705/. [Accessed: 23-Jan-2019].

- [134] “Calculation and Measurement of the Time to Complete In-Vehicle Navigation and Route Guidance Tasks (J2365 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2365_201607. [Accessed: 23-Jan-2019].
- [135] “Blind Spot Monitoring System (BSMS): Operating Characteristics and User Interface (J2802 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/J2802_201506/. [Accessed: 11-Dec-2018].
- [136] “Human Factors Definitions for Automated Driving and Related Research Topics (J3114 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/J3114_201612/. [Accessed: 23-Jan-2019].
- [137] “Process for Comprehension Testing of In-Vehicle Symbols (J2830 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2830_201606. [Accessed: 23-Jan-2019].
- [138] “Definitions and Data Sources for the Driver Vehicle Interface (DVI) (J3077 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j3077_201512. [Accessed: 23-Jan-2019].
- [139] “Measurement of Minimum Noise Emitted by Road Vehicles (J2889/1 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2889/1_201109. [Accessed: 23-Jan-2019].
- [140] “ITS In-Vehicle Message Priority (J2395 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2395_200202. [Accessed: 23-Jan-2019].
- [141] “Development of Design and Engineering Recommendations for In-Vehicle Alphanumeric Messages (J2831 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2831_201204. [Accessed: 23-Jan-2019].
- [142] “Operational Definitions of Driving Performance Measures and Statistics (J2944 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2944_201506. [Accessed: 23-Jan-2019].
- [143] “Navigation and Route Guidance Function Accessibility While Driving Rationale (J2678 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/J2678_201609/. [Accessed: 23-Jan-2019].

- [144] “Guidelines for Speech Input and Audible Output in a Driver Vehicle Interface (J2988 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2988_201506. [Accessed: 23-Jan-2019].
- [145] “Human Factors in Forward Collision Warning Systems: Operating Characteristics and User Interface Requirements (J2400 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2400_200308. [Accessed: 23-Jan-2019].
- [146] “Adaptive Cruise Control (ACC) Operating Characteristics and User Interface (J2399 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2399_201409. [Accessed: 23-Jan-2019].
- [147] “Definition of Road Vehicle Hands-Free Operation of a Person-to-Person Wireless Communication System or Device (J2972 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2972_201403. [Accessed: 23-Jan-2019].
- [148] “Vulnerable Road User Safety Message Minimum Performance Requirements (J2945/9 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2945/9_201703. [Accessed: 23-Jan-2019].
- [149] “Dedicated Short Range Communications (DSRC) Message Set Dictionary™ ASN file (J2735ASN Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2735asn_201603. [Accessed: 23-Jan-2019].
- [150] “Dedicated Short Range Communications (DSRC) Message Set Dictionary (J2735 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2735_201509. [Accessed: 23-Jan-2019].
- [151] “On-Board System Requirements for V2V Safety Communications (J2945/1 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2945/1_201603. [Accessed: 23-Jan-2019].
- [152] “Dedicated Short Range Communications (DSRC) Performance Requirements for V2V Safety Awareness (J2945/2 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j2945/2_201810. [Accessed: 23-Jan-2019].

- [153] “Candidate Improvements to Dedicated Short Range Communications (DSRC) Message Set Dictionary [SAE J2735] Using Systems Engineering Methods (J3067 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j3067_201408. [Accessed: 23-Jan-2019].
- [154] “Dynamic Test Procedures for Verification & Validation of Automated Driving Systems (ADS) (WIP = Work In Progress).” [Online]. Available: <https://www.sae.org/works/documentHome.do?docID=J3092&inputPage=wIpSdOcDeTallS&comtID=TEVAVS>. [Accessed: 23-Jan-2019].
- [155] “Guidelines for Safe On-Road Testing of SAE Level 3, 4, and 5 Prototype Automated Driving Systems (ADS) (J3018 Ground Vehicle Standard) - SAE Mobilus.” [Online]. Available: https://saemobilus.sae.org/content/j3018_201503. [Accessed: 23-Jan-2019].
- [156] “Preparing for the Future of Transportation - Automated Vehicles 3.0 -,” U.S. Department of Transportation, Oct. 2018.
- [157] “Test de systèmes embarqués en conditions réelles (FOT) | Mobilité Intelligente.” [Online]. Available: <https://www.mobilite-intelligente.com/article/annuaire/ecosysteme/recherche-developpement/Projets-de-recherche/test-de-systemes-embarques-en>. [Accessed: 22-Jan-2019].
- [158] “Volvo Intelligent Vehicle Initiative Field Operational Test - FOT-Net WIKI.” [Online]. Available: http://wiki.fot-net.eu/index.php/Volvo_Intelligent_Vehicle_Initiative_Field_Operational_Test. [Accessed: 22-Jan-2019].
- [159] “Mack Intelligent Vehicle Initiative Field Operational Test - FOT-Net WIKI.” [Online]. Available: http://wiki.fot-net.eu/index.php/Mack_Intelligent_Vehicle_Initiative_Field_Operational_Test. [Accessed: 22-Jan-2019].
- [160] “Safety Pilot Model Deployment Data - Research Data Exchange Release 2.3 -,” U.S. Department of Transportation, Aug. 2013.
- [161] “RDE Home Page.” [Online]. Available: <http://www.its-rde.net/>. [Accessed: 22-Jan-2019].
- [162] Booz, Allen, and Hamilton, “Safety Pilot Model Deployment - Sample Data Environment - Data Handbook V1.3 -,” US Department of Transportation Research and

Technology Innovation Administration Intelligent Transportation Systems --Joint Program Office, Dec. 2015.

- [163] “About Our Data | Weather Underground.” [Online]. Available: <https://www.wunderground.com/about/data>. [Accessed: 23-Jan-2019].
- [164] “NCEP Meteorological Assimilation Data Ingest System (MADIS).” [Online]. Available: <https://madis.ncep.noaa.gov/>. [Accessed: 23-Jan-2019].
- [165] J. Wenger, “Automotive radar - status and perspectives,” in *IEEE Compound Semiconductor Integrated Circuit Symposium, 2005. CSIC '05.*, Palm Springs, CA, USA, 2005, p. 4 pp.
- [166] W. R. Gilks, S. Richardson, D. Spiegelhalter, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, 1995.
- [167] K. P. Murphy, “Monte Carlo inference,” in *Machine Learning: A Probabilistic Perspective*, London: The MIT Press, 2012, pp. 817–837.
- [168] K. P. Murphy, “Markov chain Monte Carlo (MCMC) inference,” in *Machine Learning: A Probabilistic Perspective*, London: The MIT Press, 2012, pp. 839–876.
- [169] L. Martino, “A review of multiple try MCMC algorithms for signal processing,” *Digital Signal Processing*, vol. 75, pp. 134–152, Apr. 2018.
- [170] T. Ghirmai, “Applying Metropolis-Hastings-within-Gibbs algorithms for data detection in relay-based communication systems,” in *2015 IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE)*, Salt Lake City, UT, USA, 2015, pp. 167–171.
- [171] J. Scholliers *et al.*, “D16.3 Proposal of procedures for assessment of preventive and active safety functions,” TNO Defensie en Veiligheid, PR16100SPD070831v10D16-3, Jan. 2007.
- [172] “AEB Test Procedures Autonomus Emergency Braking.” Thatcham.
- [173] T. Kim, T. Lee, K. Yi, and J. Lee, “Safety Assessment Procedure for Advanced Emergency Braking System,” presented at the 24th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration, 2015.
- [174] R. Shram, A. Williams, and A. Van Ratingen, “Safety assessment procedure for advanced emergency braking system,” Euro NCAP P-NCAP Working Group, 2015.

- [175] K. P. Murphy, “Kernels,” in *Machine learning: a probabilistic perspective*, London: The MIT Press, 2012, pp. 481–515.
- [176] C. M. Bishop, “Mixture models and EM,” in *Pattern recognition and machine learning*, Springer Science+Business Media, LLC, Singapore, 2006, pp. 423–455.
- [177] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, “Accelerated Evaluation of Automated Vehicles in Car-Following Maneuvers,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 733–744, Mar. 2018.
- [178] E. Meloche, *Autonomous Emergency Braking (AEB): Test Procedure*. 2016.
- [179] *Crash Imminent Brake System Performance Evaluation for the New Car Assessment Program*. 2015.
- [180] *Intelligent Transport Systems – Forward Vehicle Collision Mitigation Systems – Operation, Performance, and Verification Requirements*. 2012.
- [181] *European New Car Assessment Programme, Test Protocol – AEB systems*. 2015.
- [182] D. B. Suits, “Use of Dummy Variables in Regression Equations,” *Journal of the American Statistical Association*, vol. 52, no. 280, pp. 548–551, Dec. 1957.
- [183] B. S. Everitt and A. SKRONDAL, “The Cambridge Dictionary of Statistics,” 2010.
- [184] J. Vanschoren, *OpenML: exploring machine learning better, together*. 2018.
- [185] K. . Murphy, “Bayesian statistics,” in *Machine Learning : A Probabilistic Perspective*, London, England: The MIT Press, 2012, pp. 180–183.
- [186] R. Tibshirani, T. Hastie, and J. Friedman, “Model Assessment and Selection,” in *Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2017, pp. 221–260.
- [187] “Weather Montreal - meteoblue.” [Online]. Available: https://www.meteoblue.com/en/weather/forecast/week/montreal_canada_6077243. [Accessed: 05-Nov-2018].
- [188] A. A. Johnson, G. L. Jones, and R. C. Neath, “Component-Wise Markov Chain Monte Carlo: Uniform and Geometric Ergodicity under Mixing and Composition,” *Statistical Science*, vol. 28, no. 3, pp. 360–375, Aug. 2013.
- [189] *TEST PROTOCOL - AEB systems*. 2017.

- [190] D. Pelleg and A. W. Moore, “X-means: Extending K-means with Efficient Estimation of the Number of Clusters,” in *Proceedings of the Seventeenth International Conference on Machine Learning*, San Francisco, CA, USA, 2000, pp. 727–734.
- [191] “Cloud Application Platform | Heroku.” [Online]. Available: <https://www.heroku.com/>. [Accessed: 24-Jan-2019].
- [192] “Prediction [APP1] | AEB System | PMG Technologies.” [Online]. Available: <https://liv-pmg-app1-nc.herokuapp.com/>. [Accessed: 24-Jan-2019].
- [193] “Classification APP-2 | AEB System | PMG Technologies.” [Online]. Available: <https://liv-pmg-app2-nc.herokuapp.com/>. [Accessed: 24-Jan-2019].

LISTE DES PUBLICATIONS

1. N. E. Chelbi, D. Gingras, and C. Sauvageau, “New Field Operational Tests Sampling Strategy Based on Metropolis-Hastings Algorithm,” in *Intelligent Systems and Applications. IntelliSys 2018. Advances in Intelligent Systems and Computing*, 2019, vol. 868. (Publié)
2. N. E. Chelbi, D. Gingras, and C. Sauvageau, “Proposal of a New Virtual Evaluation Approach of Preventive Safety Applications and Advanced Driver Assistance Functions Based on Metropolis Hastings Algorithm - Application: AEB System -,” *Proceedings, ITS World Congress (ITSWC 2018)*, Copenhagen, Denmark, Nov. 2018. (Publié)
3. N. E. Chelbi, D. Gingras, and C. Sauvageau, “Modified Metropolis-Hastings Algorithm for Efficient Sampling from Discrete Probability Distributions (MMH-DPD) Applied to Field Operational Tests database (SPMD),” in *IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, 2018, pp. 125–131. (Publié)
4. N. E. Chelbi, D. Gingras, and C. Sauvageau, “Proposal of a new virtual evaluation approach of preventive safety applications and advanced driver assistance functions – application: AEB system,” *IET Intelligent Transport Systems*, vol. 12, no. 9, pp. 1148–1156, 2018. (Publié)
5. N. E. Chelbi, D. Gingras, and C. Sauvageau, “Worst-case Scenarios Identification Approach for the Evaluation of Advanced Driver Assistance Systems in Intelligent/Autonomous Vehicles under Multiple Conditions,” *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, p. 34. (Soumis)

LISTE DES RAPPORTS

1. Les véhicules connectés - Préparé par : Nacer Eddine Chelbi - (Document interne - PMG Technologies).
2. Centre d'essais et de recherche de la mobilité intelligente et de l'innovation (CERMII) - Préparé par : Nacer Eddine Chelbi et Ayet Bagane - (Document interne - PMG Technologies).
3. Étude de l'état de l'art en homologation, évaluation et essais des véhicules intelligents (actuel et au niveau mondial) - Préparé par : Nacer Eddine Chelbi - (Document interne - PMG Technologies)
4. Proposition des scénarios envisageables pour les essais sur piste des véhicules intelligents - Préparé par : Nacer Eddine Chelbi - (Document interne - PMG Technologies)
5. Étude des plateformes matérielles pouvant être utilisées pour le développement d'un protocole d'essais étendu impliquant des essais physiques et virtuels - Préparé par : Nacer Eddine Chelbi - (Document interne – PMG Technologies)
6. Proposition d'une plateforme de test et validation MIL/SIL/PIL/HIL - Préparé par : Nacer Eddine Chelbi - (Document interne - PMG Technologies)